软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2024,35(1):185−219 [doi: 10.13328/j.cnki.jos.006834]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

图像对抗样本检测综述*

周 涛[1,2], 甘 燃[1,2], 徐东伟[1,2], 王竟亦[3], 宣 琦[1,2]

[1](浙江工业大学 网络空间安全研究院, 浙江 杭州 310023)

[2](浙江工业大学 信息工程学院, 浙江 杭州 310023)

[3](浙江大学 控制科学与工程学院, 浙江 杭州 310058)

通信作者: 宣琦, E-mail: xuanqi@zjut.edu.cn

摘 要: 深度神经网络是人工智能领域的一项重要技术, 它被广泛应用于各种图像分类任务. 但是, 现有的研究表明深度神经网络存在安全漏洞, 容易受到对抗样本的攻击, 而目前并没有研究针对图像对抗样本检测进行体系化分析. 为了提高深度神经网络的安全性, 针对现有的研究工作, 全面地介绍图像分类领域的对抗样本检测方法. 首先根据检测器的构建方式将检测方法分为有监督检测与无监督检测, 然后根据其检测原理进行子类划分. 最后总结对抗样本检测领域存在的问题, 在泛化性和轻量化等方面提出建议与展望, 旨在为人工智能安全研究提供帮助.

关键词: 深度神经网络; 对抗样本检测; 人工智能安全; 图像分类

中图法分类号: TP391

中文引用格式: 周涛, 甘燃, 徐东伟, 王竟亦, 宣琦. 图像对抗样本检测综述. 软件学报, 2024, 35(1): 185–219. http://www.jos.org.cn/1000-9825/6834.htm

英文引用格式: Zhou T, Gan R, Xu DW, Wang JY, Xuan Q. Survey on Adversarial Example Detection of Images. Ruan Jian Xue Bao/Journal of Software, 2024, 35(1): 185–219 (in Chinese). http://www.jos.org.cn/1000-9825/6834.htm

Survey on Adversarial Example Detection of Images

ZHOU Tao[1,2], GAN Ran[1,2], XU Dong-Wei[1,2], WANG Jing-Yi[3], XUAN Qi[1,2]

[1](Institute of Cyberspace Security, Zhejiang University of Technology, Hangzhou 310023, China)

[2](College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China)

[3](College of Control Science and Engineering, Zhejiang University, Hangzhou 310058, China)

Abstract: As an important technology in the field of artificial intelligence (AI), deep neural networks are widely used in various image classification tasks. However, existing studies have shown that deep neural networks have security vulnerabilities and are vulnerable to adversarial examples. At present, there is no research on the systematic analysis of adversarial example detection of images. To improve the security of deep neural networks, this study, based on the existing research work, comprehensively introduces adversarial example detection methods in the field of image classification. First, the detection methods are divided into supervised detection and unsupervised detection by the construction method of the detector, which are then classified into subclasses according to detection principles. Finally, the study summarizes the problems in adversarial example detection and provides suggestions and an outlook in terms of generalization and lightweight, aiming to assist in AI security research.

Key words: deep neural network (DNN); adversarial example detection; AI security; image classification

近年来, 深度学习, 特别是深度神经网络 (deep neural network, DNN), 在图像分类[1,2]、语音识别[3,4]、信号分析[5,6]、图数据挖掘[7,8]等领域取得了巨大的成功. 以深度学习为代表的人工智能技术引起了学术界和工业界的高度重视, 掀起了新一轮的人工智能热潮. 然而, 深度神经网络强大性能的背后, 依然存在不可忽视的缺陷, 研究表

* 基金项目: 浙江省重点研发计划 (2022C01018); 国家自然科学基金 (U21B2001, 62102359)
收稿时间: 2022-01-24; 修改时间: 2022-10-21; 采用时间: 2022-11-17; jos 在线出版时间: 2023-08-09
CNKI 网络首发时间: 2023-08-10

明, 它容易受到对抗样本的攻击: 攻击者只需在原输入上添加精心设计的细微扰动, 即可使深度神经网络的决策产生错误.

由于人工智能技术的普及, 深度神经网络被广泛地应用于各种真实场景, 例如自动驾驶, 因而其安全性在未来的研究中将显得尤为重要. Eykholt 等人[9]开发了一种名为鲁棒物理扰动 (robust physical perturbations, RP2) 的算法, 它可以生成一些黑白或彩色的图像, 通过在路牌上贴上这些图案, 自动驾驶汽车中用于记录和分析道路信息的路标分类器就会被这些图案所愚弄, 进而导致分类出错, 而这极有可能造成重大的安全事故. 由此可见, 对抗样本的存在将会极大地影响深度学习算法的进一步应用, 检测和防御对抗样本将成为保障深度神经网络安全性、可靠性的关键技术之一.

目前, 对抗样本防御技术尝试使用各种方法来强化 DNN, 例如对抗训练[10]、梯度掩盖[11]等, 前者将对抗样本加入到训练集, 训练加强模型, 但是需要具备攻击的先验知识; 后者旨在掩盖梯度, 通过以较小的梯度训练模型从而增强训练过程, 使模型对输入的微小变化不敏感. 但是, 实验表明, 这可能会导致对正常样本的分类精度下降. Papernot 等人[12]介绍了一种称为蒸馏防御的技术. 该技术分别训练两个网络, 其中第一个网络生成概率向量来标记原始数据集, 而另一个网络则使用新标记的数据集进行训练, 这可以降低模型对扰动的敏感程度, 提升对对抗样本的鲁棒性. 但是这些防御方法可以被经过针对性设计的攻击所击破: Carlini 等人[13]对他们的攻击方法进行参数微调便可突破防御. 同时, 对已经应用的模型进行对抗训练或者改变模型结构将带来昂贵的重训练成本, 并且面对不断更新进步的对抗攻击方法, 模型需要频繁地迭代更新. 此外, 大多数现有的防御技术均针对特定模型, 若要将防御技术应用到其他模型时, 需要重新训练, 且迁移效果不好.

为了应对上述挑战, 越来越多的研究开始聚焦于对抗样本检测, 这些方法大多不需要重新训练模型, 可以大大降低工程的复杂性. 对抗样本检测通过研究对抗性扰动的特点及其与正常样本的统计学差异, 在运行 DNN 模型时将对抗样本区分出来, 防止模型遭受到对抗样本的攻击. 同时许多检测方法与原始 DNN 模型相互独立, 具有较好的可移植性.

本文重点研究图像分类领域的对抗样本检测算法, 对现有的研究工作中较为典型或效果显著的方法进行详细的原理介绍和分析. 本文第 1 节介绍图像对抗样本检测领域的相关工作. 第 2 节介绍本文的研究背景以及相关工作, 对文中涉及的符号和概念进行说明. 第 3 节介绍各种图像分类对抗检测算法的原理、表现和优缺点. 第 4 节讨论对抗样本检测技术面临的挑战与展望. 第 5 节对本文的研究进行总结.

1 相关工作

Akhtar 等人[14]发表了涵盖对抗样本检测领域的第一篇综述, 其中主要包括了 2018 年之前完成的工作. 他们根据检测方法应用位置进行分类: 1) 改变输入数据; 2) 改变模型; 3) 添加附加网络. Wang 等人[15]对防御方法进行了总结, 并根据针对对抗样本的行为类型对防御方法进行了分类: 1) 反应式方法; 2) 主动型方法. 他们将检测算法归类为反应式. 在文献 [16] 中, 作者第一个将检测方法进行分类: 1) 辅助模型, 利用子网络或独立网络作为分类器来预测对抗样本; 2) 统计模型, 寻找特征从统计分析的角度区分正常和对抗样本; 3) 基于预测一致性的模型, 改变输入或模型参数, 判断模型预测变化. Bulusu 等人[17]和 Miller 等人[18]根据对抗样本是否参与检测器的训练过程, 将检测方法分为: 1) 有监督检测, 对抗样本参与检测器的构建或训练; 2) 无监督检测, 仅使用正常样本构建或训练检测器. Carlini 等人[19]对 10 个检测方法进行了实验研究, 他们的工作表明许多检测方法并没有达到他们声称的效果, 可以通过修改攻击参数或针对性的设计损失函数实现破解.

上述的一些文献做了大量工作, 但没有对检测方法进行更详细的分类和讨论, 并且在这之后许多研究者还发表了很多新的检测方法.

2 背景介绍

在本节中我们简要地介绍了针对图像分类深度学习模型的基本概念, 对文中所使用的符号进行说明. 其次介绍了深度学习模型面临的威胁模型. 然后描述了较为先进主流的攻击方法, 最后对常用的数据集进行了介绍, 在

第 3 节中会更详细讨论的对抗样本检测方法.

2.1 符号和定义

深度神经网络从原理上可以认为是一个拟合的函数 f, 它使用其神经节点互连从标记的原始数据中提取特征. 设 $\mathcal{X}$ 为输入空间, 例如图像样本, $\mathcal{Y}$ 为标签空间, 例如分类标签. 设 $\mathbb{P}(X,Y)$ 是 $\mathcal{X}\times\mathcal{Y}$ 上的数据分布, 其中 X 代表输入, 例如单张图像, Y 代表输出, 例如图像的标签. $f: X\to Y$ 称为预测函数. 因为, 神经网络通常使用随机梯度下降法 (stochastic gradient descent, SGD) 进行训练, 该算法使用误差的反向传播来更新模型权重 θ.

对抗样本这个概念最早由 Szegedy 等人[20]提出, 通过向输入样本 x 中添加细微的扰动得到新的输入 x', 可以使 DNN 模型以高置信度给出一个错误的输出, 这里的 x' 就称为对抗样本. 对抗样本 x' 满足 $f(x)\neq f(x')$, 并且 $\|x'-x\|<\epsilon$, 其中 ϵ 是最大允许扰动, ϵ 满足 $\epsilon\in\mathcal{R}^n$, 其中, n 表示 n 维向量.

本文涉及的一些符号及概念描述见表 1、表 2.

表 1 符号表

符号	描述
x	正常输入图像 (正常样本)
y	正常输入图像的标签
x'	x 的对抗样本
y'	对抗样本 x' 的标签
t	对抗攻击的目标标签
$Z(\cdot)$	神经网络最后的全连接层的输出, 也称Logits
$f(\cdot)$	预测函数它返回每个类别的预测概率, $f(x)=Softmax(z(x))$
θ	模型 f 的参数/权重
$\ell(,)$	损失函数
$\delta=x'-x$	扰动, 添加到正常样本上的噪声或对抗样本于正常样本之间的差异
ϵ	最大允许扰动
σ	激活函数
$\|\cdot\|_p$	ℓ_p 范数
X	训练集

表 2 定义表

定义	描述
对抗攻击	生成对抗样本的算法
防御	使深度学习模型能够抵御对抗攻击产生的对抗样本的技术
检测	能够预测出输入是否为对抗样本的技术
训练集	用于模型拟合的数据样本
验证集	模型训练过程中单独留出的样本集, 它可以用于调整模型的超参数和用于对模型的能力进行初步评估. 通常用来在模型迭代训练时, 用以验证当前模型泛化能力 (准确率, 召回率等), 以决定是否停止继续训练
测试集	用于评估模型的性能. 但不能作为调参、选择特征等算法相关的选择的依据
可迁移性	对抗样本的一种特性, 表示它迁移至其他模型的攻击能力
白盒攻击	攻击者能够获取被攻击模型的所有信息 (梯度、损失、模型结构等)
黑盒攻击	攻击者无法获取被攻击模型的信息, 只能获取模型的输入和输出
目标攻击	诱导被攻击模型将对抗样本分类为特定的目标标签
无目标攻击	诱导被攻击模型将对抗样本分类为某个任意标签但不等于原标签

2.2 对抗攻击

在本节中, 我们将介绍目前主流的一些白盒对抗攻击算法, 也是目前大部分检测算法用于评估算法效果的常

用基准算法.

● L-BFGS 攻击[20]. 它是最早提出的对抗攻击算法, 这个算法的目标在于找到最小扰动 δ, 使得添加扰动后的对抗样本的预测标签 $y'=t$, 并且 x' 处于输入域的范围内. 通过引入损失函数 ℓ 转化为该优化问题:

$$\underset{\delta}{\operatorname{argmin}}\, c\,\|\delta\|+\ell(x',t) \tag{1}$$

其中, c 是不断优化寻找的最小 δ 的常量, 对于每一个常量 $c>0$, 不断优化求解这个最小问题, 最终每一个 c 都能找到一个满足问题的解, 通过执行全局的线性搜索, 最终找到扰动最小的对抗样本.

● FGSM 攻击[10]. 这是第 1 种使用深度学习模型梯度生成对抗样本的 L_∞ 距离攻击. FGSM 攻击是一种一步式的梯度更新算法, 它在输入 x 的每个像素寻找扰动方向, 即梯度符号, 从而使模型的损失最大化, 具体表示为:

$$x'=x+\epsilon\operatorname{sign}(\nabla_x\ell(x,y)) \tag{2}$$

其中, ϵ 是最大可允许的扰动, 满足 $\|x'-x\|_\infty<\epsilon$.

● BIM 攻击[21]. 它是 FGSM 攻击的迭代版本. BIM 通过执行 k 次 FGSM, 并在每一步之后都进行修剪得到结果的像素值, 保证结果处于输入域的范围内. BIM 具体表示为:

$$x'_{i+1}=x'_i+\alpha\operatorname{sign}(\nabla_x\ell(x'_i,y)),\ x'_0=x,\ i\in[0,k] \tag{3}$$

其中, α 表示控制第 i 次迭代的步长参数, 并且 $0<\alpha<\epsilon$.

● PGD 攻击[22]. 它是一种类似 BIM 的迭代攻击方法, 为了找到模型的局部最大损失, PGD 攻击每次从输入样本周围的 L_p-ball 的随机扰动开始, 每次迭代都会将扰动投射到规定范围内.

● CW 攻击[23]. Carlini 等人沿用了 L-BFGS 的优化问题 (见公式 (1)), 他们使用一个新的目标函数代替了原来的损失函数:

$$g(x')=\max\left(\max_{i\neq t}(Z(x')_i)-Z(x')_t,-k\right) \tag{4}$$

其中, Z 是 *Softmax* 函数, k 是置信度参数. 整个优化问题就变为:

$$\min_{\delta}\|\delta\|+cg(x') \tag{5}$$

$$\delta=\frac{1}{2}(\tanh(w)+1)-x \tag{6}$$

对于目标类别 t, 最小化目标函数 g 有助于帮我们找到具有更高置信度的 x'. 作者通过引入 w 来控制输入样本的扰动, 经优化从框约束问题转换为无约束问题. 并且 CW 攻击带有 3 个变体, 用于衡量基于 L_0、L_2、L_∞ 距离的 x' 与 x 的相似度.

● JSMA 攻击[24]. 它是一种使用雅可比显著图来生成对抗样本的方法. JSMA 利用深度学习模型的梯度构建显著性矩阵, 该矩阵对图像中每个像素的重要性进行排序, 通过算法选择显著性最大的像素, 并对其进行修改, 以增加分类为目标标签 t 的概率. 重复这一过程, 直到模型对输入样本的预测标签变为 t.

● DeepFool 攻击[25]. 它可以在产生比 FGSM 更小的扰动的情况下, 实现攻击. 给定一个二元仿射分类器 $\mathcal{F}=\{x:f(x)=0\}$, 其中 $f(x)=w^{\mathrm{T}}x+b$, DeepFool 攻击将从 x 到 $\mathcal{F}$ 的正交投影定义为改变分类器预测结果所需的最小扰动, 该扰动计算公式如下:

$$\delta_*=-\frac{f(x)}{\|w\|^2}w \tag{7}$$

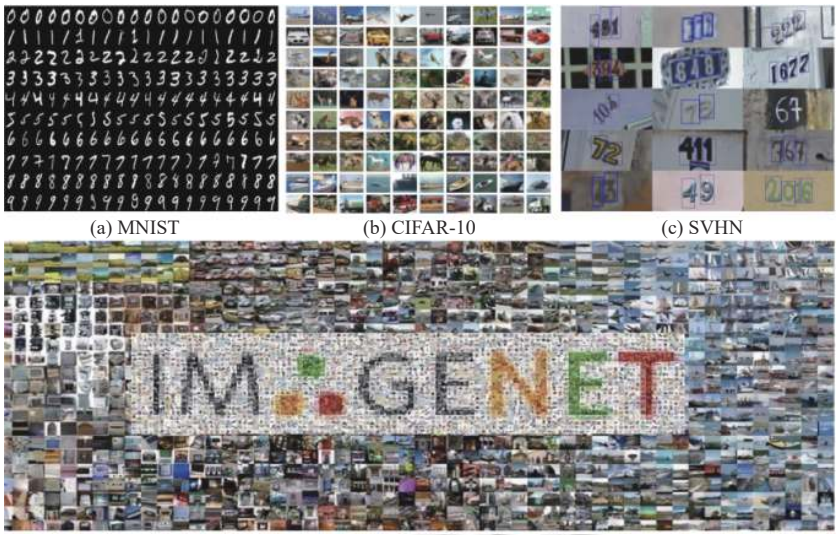
在每次迭代中, DeepFool 攻击解决如下优化问题:

$$\underset{\delta_i}{\operatorname{argmin}}\,\|\delta_i\|_2,f(x_i)+\nabla,f(x_i)^{\mathrm{T}}\delta_i=0 \tag{8}$$

最终这些扰动积累获得最终的对抗攻击扰动.

2.3 常用数据集

图 1 展示了常用的数据集.



(a) MNIST (b) CIFAR-10 (c) SVHN

(d) ImageNet

图 1 常用数据集

MNIST 手写字数据集[26], 该数据集由 250 个人的手写数字组成, 其中 50% 是高中学生, 50% 来自人口普查局的工作人员. 它的训练集包含 60 000 个样本, 测试集包含 10 000 个样本. 里面的数字由 0–9 组成, 每张图像由 28×28 个像素点组成, 每个像素点用一个灰度值表示. 几乎所有涉及图像分类的攻击、防御、检测方法, MNIST 都被用来验证算法的有效性.

CIFAR-10 数据集[27], 该数据集是一个用于物体识别的计算机视觉数据集, 它共由 60 000 张 32×32 彩色 RGB 图像组成, 总共有 10 个分类, 分别为: 飞机、汽车、鸟、猫、鹿、狗、青蛙、马、船、卡车. 每个类别有 6 000 张图片, 其中训练集与测试集的比例为 5:1.

ImageNet 数据集[28], 该数据集由斯坦福大学的李飞飞教授带领创建. 该数据集包含 14 197 122 张图片和 21 841 个 Synset 索引. Synset 是 WordNet 层次结构中的一个节点, 它又是一组同义词集合. ImageNet 数据集一直是评估图像分类算法性能的基准. ImageNet 数据集中的图片涵盖了生活中能看到的大部分事物类别, 研究者做图像分类验证时, 一般会选择其中的一些子集, 例如 Tiny-ImageNet. Tiny-ImageNet 数据集是 2016 年由斯坦福大学发布的图像分类数据集. 它是 ImageNet 的子集, 包含 200 类, 每个类有 500 张训练图片, 50 张验证图片, 50 张测试图片.

SVHN 数据集[29], 该数据集来源于谷歌街景门牌号码, 数据集中的每张图片是带有字符级边界框的彩色门牌号图像, 类似 MNIST 的 32×32 以单个字符为中心的图像, 总共包含 0–9 这 10 个类别. 它的训练集包含 73 257 个样本, 测试集包含 26 032 个样本, 以及 531 131 个额外的、难度稍低的样本, 用作额外的训练数据.

3 对抗样本检测方法

3.1 检测算法分类

本节总结了深度神经网络对抗样本检测的相关研究, 总计 24 种对抗样本检测算法, 所有的检测算法最终都能被抽象成一个检测器, 该检测器依据阈值、分类网络、预测不一致等方式判断输入样本是否为对抗样本. 为了帮助读者更加清晰地了解对抗样本检测领域的相关算法原理与研究现状, 本文从原理层面对所有检测方法进行了总结和分类, 如图 2 所示. 首先根据检测算法在设计和实现检测器时是否使用对抗样本, 将检测算法分为两个大类: 无监督检测和有监督检测.

在无监督检测中, 检测算法仅使用正常样本设计并训练检测器, 无监督检测的目标主要是减少攻击者可用的输入空间. 无监督检测根据检测原理可分为统计方法、降噪、生成式对抗网络、神经网络特性和特征对齐 5 个子类.

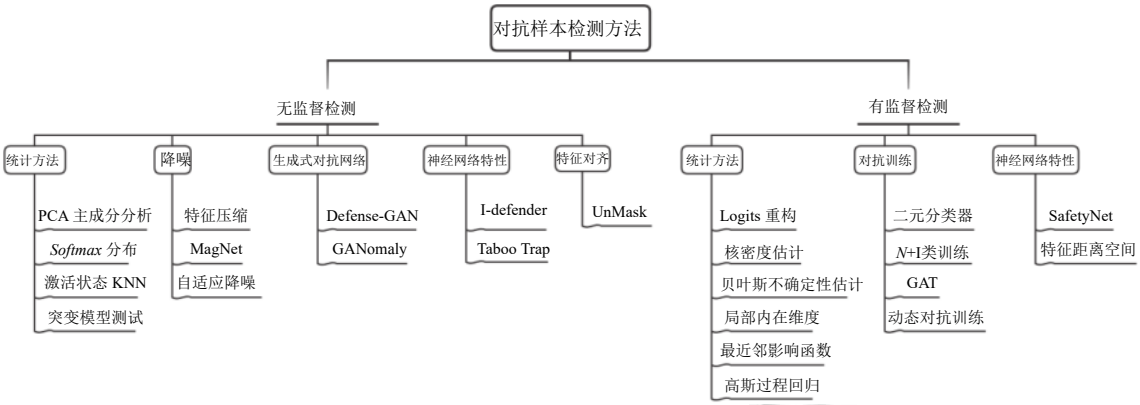


图 2 对抗样本检测方法分类

1) 统计方法, 这类检测算法探索了样本在不同层面的统计特征, 用于构建检测器. Hendrycks 等人[30]通过统计分析发现正常样本与对抗样本在其主成分白化输入系数的方差和 *Softmax* 分布上存在差异性. Carrara 等人[31]利用待测样本在 DNN 中间某层的激活状态与最近邻样本的激活状态的评分检测对抗样本. Wang 等人[32]利用标签变化率来衡量样本在突变模型中输出的标签以及原始模型输出的标签的一致性. 这些检测算法都从统计层面出发, 启发式地从大量正常样本与对抗样本中找出差异, 最后根据正常样本的表现情况, 通过设定阈值的形式实现检测. 此外, 该类检测算法所利用的属性通常与样本自身的分布/数据特点相关, 例如 PCA 主成分分析通过各种降维方法得出正常样本与对抗样本的差异部分; *Softmax* 分布与突变模型测试都利用了对抗样本处于 DNN 决策边界面附近的特点, 该特点导致其概率分布趋于平均、标签变化率更高; 激活状态最近邻本质上也是利用了待测样本与正常样本分布的相近程度进行评分的.

2) 降噪, 这类检测算法通过对输入样本进行类似降噪的处理, 探索降噪后正常样本与对抗样本之间的差异, 因为对抗样本本质上可视为一种加了特殊噪声的正常样本, 最后通过阈值或预测不一致的形式实现检测. Xu 等人[33]比较了正常样本与对抗样本在通过压缩器前后的预测差异大小, 发现对抗样本通常有更大的差异. Meng 等人[34]提出的 MagNet 检测框架训练了一个降噪自编码器用于重构输入, 并且发现对抗样本与正常样本相比会存在更高的重构误差. Liang 等人[35]引入了标量量化和平滑空间滤波来降低噪声, 同时利用图像的熵作为度量指标从而对不同类型的图像进行自适应降噪, 最终通过降噪前后样本的预测标签变化情况进行判断. 基于降噪的方法一个比较明显的特征是使用了降噪器, 但是降噪器无法保证消除所有的噪声, 并且有可能会造成失真, 导致原始模型的精度下降.

3) 生成式对抗网络 (generative adversarial network, GAN) 是一种无监督学习框架, 通过框架中的生成模型 (generative model) 和判别模型 (discriminative model) 互相博弈学习产生符合要求的输出[36]. 这类检测算法引入了 GAN 的思想, 通过构建一个对抗式的环境优化降噪器、检测器等的训练过程实现检测. Samangouei 等人[37]提出了 Defense-GAN 检测框架, 同时训练了一个生成器模型和一个检测器模型, 鼓励生成器生成接近训练数据 (正常样本) 的样本, 使对抗样本远离生成器的生成范围. Akcay 等人[38]提出了 GANomaly 检测框架, 在一个对抗式的环境中, 训练了两个解码器、一个编码器和一个检测器. 这两种检测算法最终都通过设定阈值的形式实现检测.

4) 神经网络特性, 这类检测算法与基于统计方法的检测算法研究思路的出发点不同, 统计方法相关的检测算法探寻样本本身的属性、特质, 而基于神经网络特性的检测算法主要通过观察样本在神经网络中的表现或行为构建检测器, 更多利用到了神经网络本身的一些特性. Zheng 等人[39]利用高斯混合模型对 DNN 隐藏神经元的输出进行近似计算, 并将结果作为特征值以阈值形式构建检测器实现检测. Shumailov 等人[40]对 DNN 各层的激活值进行分析, 设计转换函数接受激活值并参与模型的重训练, 将正常样本的激活值限制在特定的区间内, 并将该区间作为阈值实现检测. 这两种检测算法都在不同程度上利用了神经网络的内部神经元的输出值、激活值特性.

5) 特征对齐, 这类检测算法的主要代表是 Freitas 等人[41]提出的 UnMask 检测框架, UnMask 从图像样本中提取鲁棒特征并将其与预测标签的预期特征进行比较, 计算相似度评分, 并设定阈值实现检测. 这类检测算法的特点

决定了它只能处理特定的图像分类任务, 因为其检测所需的特征需要预先进行标注, 与数据集强相关, 可迁移性较差.

在有监督检测中, 正常样本与对抗样本都会参与检测器的设计与训练, 有监督检测方法的主要局限在于它们需要有关对抗样本的先验知识, 对未知的攻击的泛化能力较差. 有监督检测根据检测原理可分为统计方法、对抗训练和神经网络特性 3 个子类.

1) 统计方法, 这类检测算法同样探索了正常样本与对抗样本在数据分布等方面的统计特征, 但在构建检测器时, 同时使用了正常样本与对抗样本的特征. Hendrycks 和 Gimpel[30]增加了一个解码器重构 Logits, 发现对抗样本与重构误差比正常样本大, 因此将其和正常输出的 Logits 与置信度一起输入至检测器训练. Feinman 等人[42]探索了正常样本与对抗样本的数据分布子空间的差异, 使用核密度估计检测远离类流形的对抗样本, 使用贝叶斯不确定性估计检测靠近类流形的对抗样本. Ma 等人[43]则使用局部内在维度计算待测样本与其邻居的距离分布, 以评估该待测样本周围区域的空间填充能力. Cohen 等人[44]设计了一个最近邻影响函数, 衡量样本的有益样本与其最近邻训练样本的关联性并作为特征, 使用正常样本与对抗样本训练检测器实现检测. Lee 等人[45]将样本的 Logits 作为高斯过程回归模型的观察数据, 然后拟合观测数据, 根据拟合结果检测. 上述 6 种检测算法都在不同程度上探索了样本数据本身分布差异, 核密度估计、贝叶斯不确定性估计、局部内在维度和最近邻影响函数都较为明显地利用了样本之间的相对位置关系; Logits 重构和高斯回归模型利用样本在神经网络中的高维特征 Logits, 实现了样本分布差异的区分.

2) 对抗训练, 这类检测算法最大的特点是他们没有尝试直接去寻找正常样本与对抗样本之间的差异, 而是引入了对抗训练的思想, 通过构建检测神经网络, 将正常样本与对抗样本直接或间接输入至神经网络训练, 利用神经网络的强大学习能力学习正常样本与对抗样本之间的差异, 从而实现检测对抗样本的目的. Grosse 等人[46]提出了一种对抗训练的变体, 在原始分类网络中增加一个 N+1 类作为对抗样本的类别, 输入正常样本与对抗样本重新训练分类网络, 待测样本被预测为 N+1 类则认为是对抗样本. Gong 等人[47]直接构建了一个二分类检测网络, 将正常样本与对抗样本输入至二分类检测网络进行训练. Yin 等人[48]为每一个类别单独构建二分类检测网络, 输入该类型的正常样本与对抗样本进行训练. Metzen 等人[49]提出了一种动态对抗训练的检测算法, 他们将原始神经网络的多个中间层的输出作为新的检测神经网络的输入进行训练, 最后使用该检测神经网络进行检测. 在上述的 4 种检测方法中, 前 3 种都将样本直接输入至检测神经网络训练, 而第 4 种动态对抗训练算法则是利用样本在 DNN 各中间层的输出作为输入训练检测神经网络, 但他们在本质上都是依靠检测神经网络学习差异, 而不是通过寻找到的差异进行训练.

3) 神经网络特性, 这类方法主要通过观察正常样本和对抗样本在神经网络中的表现或行为的差异性构建检测器, 更多地利用到了神经网络本身的特性. Lu 等人[50]提出了 SafetyNet 检测算法, 他们将输入样本在最后一层 ReLU 中计算得到的离散编码作为支持向量机的输入, 构建检测器检测对抗样本. Carrara 等人[51]对 DNN 特征空间中特定样本的网络内部激活位置进行编码, 提出了特征距离空间检测算法, 对抗样本的激活位置会偏离正常样本所在的参考位置, 根据输入样本的激活轨迹差异实现对抗样本的检测. 这两种检测算法都利用了神经网络的激活状态, SafetyNet 利用激活状态作为离散编码, 特征距离空间则是对样本在神经网络中神经元的激活轨迹进行编码, 两者表现形式不同, 但底层依据相同.

对检测算法进行分类, 可以为对抗样本检测领域的研究提供灵感, 挖掘不同类、相同类检测算法的异同, 加深我们对各种检测算法底层逻辑的理解, 例如汇总上述各类检测算法最终采用的检测形式后, 我们可以发现有的无监督检测算法的主要检测形式为阈值, 只有特征压缩检测算法使用预测不一致形式; 而有监督检测大部分最终都采用训练检测分类网络的检测形式, 这也与其构建检测器时所使用样本数据的差异相符合.

本文将在第 3.2 节和第 3.3 节详细介绍上述所有检测算法的详细原理及特点, 为了方便读者进行查阅, 本文将检测算法评估时所使用的攻击算法和检测效果汇总成表 3. 因为不同检测算法所使用的模型、实验设置、对抗样本类型、参数都是不同的, 如果直接使用某个或最好的检测结果数值是不公平的. 因此在检测效果栏我们使用 5 个检测准确率范围表示: A (90–100), B (80–89), C (70–79), D (50–69), E (0–49), 检测效果评估参照检测算法论文原文中实验结果计算平均值获得. 若表中出现“—”, 代表在文献原文中作者没有明确给出具体数值的实验结果.

此外, 本文还将所有检测算法按照类别、子类、原理、检测形式和特点 5 个维度进行汇总描述, 帮助读者快

速了解算法的核心, 具体如表 4 所示. 在原理栏本文对检测算法的核心原理进行概括介绍; 在检测形式栏列出检测算法最终所采用的检测形式; 在特点栏, 本文从优缺点和表现两个角度对算法的主要特点概括描述.

表 3 检测算法评估汇总

类别	子类	方法	对抗攻击算法	检测效果
无监督检测	统计方法	PCA主成分分析[30]	FGSM, BIM	MNIST (A)
		*Softmax*分布[30]	FGSM, BIM	—
		激活状态KNN[31]	FGSM, L-BFGS	ImageNet (B)
		突变模型测试[32]	FGSM, JSMA, CW, DeepFool	MNIST (A), CIFAR-10 (A)
	降噪	特征压缩[33]	FGSM, BIM, JSMA, CW, DeepFool	MNIST (A), CIFAR-10 (D), ImageNet (C)
		MagNet[34]	FGSM, JSMA, CW, DeepFool	MNIST (A), CIFAR-10 (B)
		自适应降噪[35]	FGSM, CW, DeepFool	MNITS (A), ImageNet (A)
	生成式对抗网络	Defense-GAN[37]	FGSM, CW	MNITS (A), F-MNIST (B)
		GANomaly[38]	—	—
	神经网络特性	I-defender[39]	FGSM	MNITS (A), CIFAR-10 (B)
		Taboo Trap[40]	FGSM, PGD, DeepFool	MNIST (D), CIFAR-10 (D)
	特征对齐	UnMask[41]	FGSM, PGD	UnMaskDataset (B)
有监督检测	统计方法	Logits重构[30]	FGSM, BIM	—
		核密度估计[42]	FGSM, BIM, JSMA, CW	MNIST (B), CIFAR-10 (B), SVHN (D)
		贝叶斯不确定性[42]	FGSM, BIM, JSMA, CW	MNIST (B), CIFAR-10 (D), SVHN (D)
		局部内在维度[43]	FGSM, BIM, JSMA, CW	MNIST (A), CIFAR-10 (A), SVHN (A)
		最近邻影响函数[44]	FGSM, PGD, JSMA, CW, DeepFool	CIFAR-10 (A), CIFAR-100 (A), SVHN (A)
		高斯过程回归[45]	FGSM, BIM, JSMA, CW, DeepFool	MNIST (A), CIFAR-10 (B)
	对抗训练	二元分类器[1]	FGSM, JSMA	MNIST (A), CIFAR-10 (A), SVHN (A)
		N+1类对抗训练[47]	FGSM, JSMA	MNIST (A)
		GAT[48]	PGD	MNIST (A), CIFAR-10 (A)
		动态对抗训练[49]	FGSM, BIM, DeepFool	CIFAR-10 (B)
	神经网络特性	SafetyNet[50]	FGSM, BIM, DeepFool	CIFAR-10 (B), ImageNet (C)
		特征距离空间[51]	FGSM, BIM, L-BFGS, PGD	ImageNet (B)

表 4 检测算法汇总

类别	子类	方法	原理	检测形式	特点
无监督检测	统计方法	PCA主成分分析[30]	比较对抗样本与正常样本的白化后的主成分与系数	阈值	检测框架简单, 实现难度低; 但可以被特定攻击参数规避, 仅在特定数据集上有效
		*Softmax*分布[30]	测量均匀分布和*Softmax*分布之间的K散度	阈值	检测框架简单, 效率高; 可检测的对抗样本类型单一, 无实验验证
		激活状态KNN[31]	取DNN模型中间层激活状态计算KNN评分区分对抗样本	阈值	误检率低, 执行效率高, 可作为批量筛选机制; 阈值确定需要先验知识, 可迁移性差
		突变模型测试[32]	利用大量突变模型测试样本, 统计标签变化率并设置假设检验	阈值	检测准确率高, 在各攻击算法下表现稳定; 大量突变模型耗费时空资源
	降噪	特征压缩[33]	比较压缩与未压缩输入的差值, 设定阈值区分对抗样本	阈值	组合检测效果较好, 可迁移性强; 但对特定的攻击算法效果很差
		MagNet[34]	训练降噪自编码器, 设定阈值区分对抗样本与正常样本的重构误差和概率分歧	阈值	平均检测效果好, 检测框架易于部署, 对原始模型分类精度影响小; 但容易受到替代模型攻击, 检测效果依赖原始模型
		自适应降噪[35]	根据熵值, 利用标量量化和平滑空间滤波器进行自适应降噪处理	预测不一致	不需要攻击的先验知识, 自适应调整检测策略; 依赖于原始分类器准确率, 对特定攻击效果差

表 4 检测算法汇总 (续)

类别	子类	方法	原理	检测形式	特点
无监督检测	生成式对抗网络	Defense-GAN[37]	比较输入与GAN生成器生成的样本之间的均方误差	阈值	可直接与任何分类器结合; 梯度下降计算耗时, 依赖于GAN的质量, 不够稳定
		GANomaly[38]	利用GAN学习图像和潜在的空间向量捕捉训练数据的分布	阈值	不同数据集下泛化能力强; 容易受到噪声的影响
	神经网络特性	I-defender[39]	使用分类器的内在隐态分布区分对抗样本	阈值	不同攻击参数下表现稳定, 可跨领域移植, 计算销量高; 多阈值寻优依赖训练数据
		Taboo Trap[40]	利用对抗样本前向传播中的激活值超过某一阈值这一异常现象进行对抗样本检测	阈值	检测推理速度快; 需要重训练, 检测效果依赖于转移函数, 对特定攻击算法无效
	特征对齐	UnMask[41]	判断提取的鲁棒性特征与标签真实鲁棒性特征的重叠得分	阈值	检测原理直观, 具备可解释性; 需要对数据集进行标注, 对扰动参数敏感
有监督检测	统计方法	Logits重构[30]	将辅助解码器的重构输入、Logits和置信度一起作为输入训练检测器	训练检测网络	检测框架简单易于部署; 辅助解码器训练依赖原始模型, 可被攻击破解规避
		核密度估计[42]	取最后一个隐藏层特征估算核密度	阈值/训练检测网络	可检测远离类流形的对抗样本; 可被特定攻击约束规避
		贝叶斯不确定性[42]	使用dropout衡量分类网络的贝叶斯不确定性	阈值/训练检测网络	可检测靠近类流形的对抗样本, 攻击所需扰动更大; 但需要进行多次推理
		局部内在维度[43]	估算样本在每一个转换层的局部内在维度, 训练检测器	训练检测网络	对对样本的表征能力强; 但对攻击置信度参数敏感
		最近邻影响函数[44]	衡量输入样本的有益样本与其最近邻训练样本关联性	训练检测网络	检测效果在各攻击算法下均表现稳定; 但训练及推理阶段计算量大
		高斯过程回归[45]	利用最后一层隐藏层的高维特征训练高斯回归检测器	训练检测网络	少量训练数据实现高效检测; 对部分攻击检测效果较差, 协方差选取受数据维度影响
	对抗训练	二元分类器[1]	在原分类网络上增加一类为对抗样本的类别训练网络	训练检测网络	充分利用原始分类网络性能; 不适用于复杂数据集, 假阳率过高
		N+1类对抗训练[47]	直接将对抗样本与正常样本输入检测器训练	训练检测网络	部署简单; 但对攻击扰动系数敏感, 泛化能力差
		GAT[48]	对数据集的每一类样本单独训练二分类检测器, 通过Gibbs分布设定阈值检测对抗样本	阈值	检测准确率高; 阈值确定依赖训练数据, 训练成本高
		动态对抗训练[49]	用网络中间层特征训练检测神经网络扩充分类神经网络	训练检测网络	增加了攻击者的难度; 但对某些攻击误检率高, 依赖原始模型, 可被替代攻击规避
	神经网络特性	SafetyNet[50]	量化模型的最后一个ReLU激活层并构建一个二元SVM RBF分类器	训练检测网络	灵活利用神经元激活特征, 效率高; 依赖训练集对抗样本, 对未知攻击的检测效果差
		特征距离空间[51]	对特征空间中图像特定的网络内部激活轨迹进行编码, 区分对抗样本	训练检测网络	激活轨迹具备可解释性; 对扰动小的对抗样本检测效果差, 枢纽点确定依赖训练数据

3.2 无监督检测

3.2.1 统计方法

3.2.1.1 PCA 主成分分析

主成分分析 (principal component analysis, PCA) 是一种常见的降维方法, 它的主要思想就是将原始 n 维的特征映射到目标 k 维上, 这 k 维是全新的正交特征, 也被称为主成分. 通过 PCA 降维的数据更易适用, 降低了算法的计算开销, 并且在一定程度上减少了噪声, 使结果更具解释性. 在图像处理中, 白化一般用于对过度曝光或低曝光的图片进行处理的措施, 它通过改变图像的平均像素值为 0, 单位方差为 1 实现. 是一种降低输入冗余性的方法, 白化后的输入具有特征之间相关性较低和所有特征具有相同方差的特点.

Hendrycks 等人[30]对输入进行 PCA 白化处理, 得到了一组白化向量, 白化向量的第 1 项是具有最大特征值的特征向量, 或者说主分量的系数. 排名靠后的项则是具有较小特征值的特征向量系数. 他们发现对抗样本排名靠后的主成分的方差通常比正常样本更大, 因此可以作为区分对抗样本的依据并通过设置阈值进行检测, 其检测框架如图 3 所示, 具体检测步骤如下.

1) 处理训练数据使其大小以零为中心.

2) 计算中心数据的协方差矩阵 C, 并通过 $C=U\Sigma V^{\mathrm{T}}$ 找到 C 的奇异值分解 (SVD).

3) 得到输入的 PCA 白化向量 $\Sigma^{-1/2}U^{\mathrm{T}}x$, 并计算其方差.

4) 根据方差与阈值进行比较获得检测结果, 若结果大于阈值则判断为对抗样本, 反之, 判断为正常样本.

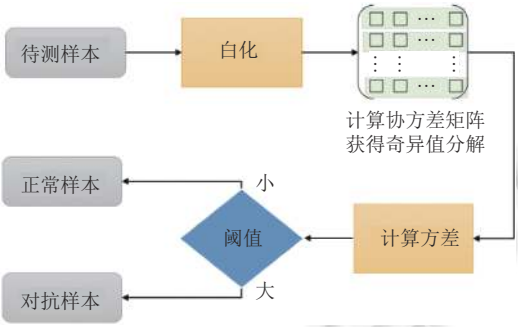


图 3 PCA 主成分分析检测框架

Hendrycks 等人使用 FGSM 和 BIM 攻击算法评估了他们的检测算法, 在 MNIST、CIFAR-10 以及 Tiny-ImageNet 数据集上均得了超过 90% 的检测准确率. 但是 Carlini 等人[23]在使用 CW 攻击评测该算法时发现, 该检测算法仅在 MNIST 数据集上有效, 在 CIFAR-10 数据集上无效. 他们进一步分析了这种现象, 认为该方法在 MNIST 上有效的原因可能是由于 MNIST 数据集中存在某些伪像. 例如, 在正常样本中, 不属于数字的像素值具有零值, 而在对抗样本中, 不属于数字的像素值通常由于对抗攻击对图像像素值的修改而不再为零, 这也解释了为什么排名靠后的主成分的方差明显较大. 因此, 如果攻击者知道该防御策略, 可以在生成对抗样本的过程中限制靠后的主要成分的变化, 绕过此检测方法.

3.2.1.2 *Softmax* 分布

Hendrycks 等人[30]提出正常样本与对抗样本之间的 *Softmax* 分布是不同的, 正常样本的 *Softmax* 向量通常拥有更大的最大概率, 因此可以用来检测对抗样本. 具体地, 他们通过测量均匀分布和 *Softmax* 分布之间的 Kullback-Leibler 散度[52], 然后根据得出的散度进行阈值检测发现: 与对抗样本相比, 正常样本的 *Softmax* 分布通常远离均匀分布, 因为模型倾向于以较高置信度预测正常样本. 该检测算法的检测框架如图 4 所示, 具体检测步骤如下.

1) 将待测样本输入至 DNN 模型, 获取其 *Softmax* 分布.

2) 计算 *Softmax* 分布向量与均匀分布之间的 KL 散度.

3) 比较 KL 散度与设定阈值间的大小, 若小于阈值, 判断为对抗样本; 大于阈值, 判断为正常样本.

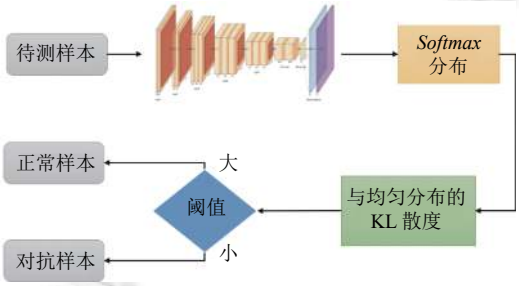


图 4 *Softmax* 分布检测框架

Hendrycks 等人在原论文中没有直接给出这种检测方法的检测实验结果, 而是尝试使用 KL 散度约束对抗样本的生成过程, 结果表明, 要获得符合正常样本 KL 散度的对抗样本, 对抗样本的图像质量必须下降, 即扰动会增大. Wiyatno 等人[53]在评估时认为这种算法只适用于攻击成功后立即停止的攻击, 例如 JSMA, 这类方法生成的对

抗样本通常具有较低的置信度, 而对高置信度类型的攻击方法并没有太大效果.

3.2.1.3 激活状态 KNN

Carrara 等人[31]对 DNN 模型中隐藏层的神经元激活状态进行了分析, 提出的了一种利用 DNN 模型中间层激活状态并结合 K 最近邻 (K-nearest neighbor, KNN) 来检测对抗样本的检测方法. 其检测框架如图 5 所示, 检测步骤如下.

1) 将待测样本输入到 DNN 中, 获得样本的预测标签并提取 DNN 模型某一中间层的激活状态.

2) 通过 KNN 算法从对应预测标签类别的训练集中找到与该层激活状态相似的训练样本.

3) 计算该样本的 KNN 评分, 若得分大于阈值, 判断为正常样本; 得分小于阈值, 判断为对抗样本.

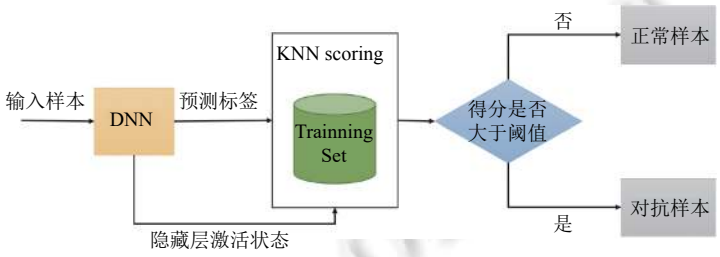


图 5 激活状态 KNN 检测框架

具体地, 假设给定输入样本 x, 其 DNN 输出标签为 t. 利用 DNN 模型中间层 (某一层) 激活状态, 从训练集中得到输入 x 样本的 k 个邻居:

$$NN(x,k)=\{(x_1,y_1),(x_2,y_2),\ldots,(x_k,y_k)\} \tag{9}$$

其中, x_i 表示第 i 个训练样本, y_i 表示该训练样本的标签, 则输入样本 x 的评分为:

$$s(x,t)=\frac{\sum_{i=1}^{k} w_i\psi\{y_i=t\}}{\sum_{i=1}^{k} w_i} \tag{10}$$

其中, w_i 是分配给对应结果的权重, $\psi\{y_i=t\}$ 的计算方式如下:

$$\psi\{t_i=t\}=\begin{cases}1, & \text{if } y_i=t\\ 0, & \text{else}\end{cases} \tag{11}$$

Carrara 等人在实际评估检测算法时对这些激活状态特征进行了处理, 在 ImageNet 数据集上进行了评估, 使用了 PCA+白化组合处理后, 在 L-BFGS 和 FGSM 攻击算法上分别取得了 86.5% 和 83.8% 的最好检测准确率. 并且 Carrara 等人在使用很低的阈值 (约 0.002) 时, 可以检测出超过 50% 由 L-BFGS 生成的对抗样本和超过 40% 由 FGSM 生成的对抗样本的情况下, 保留 98% 以上正确分类的样本, 它的误检率非常低.

最终用于检测的阈值是 Carrara 等人在权衡真阳率和假阳率在可接受范围下确定的, 但是在真实场景下, 算法设计者无法预估对抗样本在该算法下的阈值表现, 选取存在一定不合理性. 虽然激活状态 KNN 仅需提取某一层的激活状态, 但是并非所有的中间层都有很好的效果, 需要经过挑选, 因此只能针对特定模型下的特定层有效, 解释性较差, 如果更换模型或增加新数据, 需要重新训练并挑选中间层和阈值.

激活状态 KNN 这种方法虽然利用了 DNN 中间层的激活状态, 但是其分类本质还是依靠寻找最近邻、相似样本的方式, 进行评分, 更偏向于数据层面的分布, 因此将这种方法归为统计方法类. Carrara 等人更早之前提出了一种称为特征距离空间的检测方法, 它对 DNN 中的激活轨迹进行了编码, 我们会在本文中第 3.3.3.2 节详细介绍, 激活状态 KNN 实际是特征距离空间方法的一种衍生.

3.2.1.4 突变模型测试

Wang 等人[32]利用对抗样本相较于正常样本, 对于模型决策边界的改变更为敏感的特性来检测对抗样本. 将

突变检测应用于 DNN, 在保证模型准确率不受很大影响的前提下, 通过突变算子轻微改变深度神经网络内部参数或结构来达到改变模型决策边界的目的. 由于对抗样本对于决策边界的改变更加敏感, 那么生成的突变模型组输出的标签很有可能与原始模型输出的标签不一致. 突变模型检测对抗样本的流程如图 6 所示.

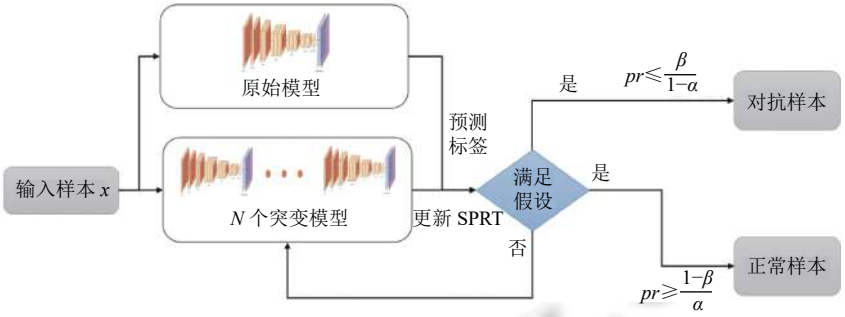


图 6 突变模型测试检测框架

他们总共使用了 4 种突变算子来生成突变模型分别为: 高斯模糊 (Gaussian fuzzing, GF): 使用高斯分布模糊权重; 权重洗牌 (weight shuffling, WS): 随机打乱选中的权重; 神经元交换 (neuron switch, NS): 交换一层中的两个神经元; 神经元激活翻转 (neuron activation inverse, NAI): 改变神经元的激活状态.

他们提出了标签变化率 (lable change rate, LCR) 这一概念, 用于衡量样本在突变模型中输出的标签以及原始模型输出的标签的一致性, 利用对抗样本在突变模型的输出标签会改变这一性质可以检测对抗样本. LCR 具体定义为:

$$\varsigma(x)=\frac{|\{m_i \mid m_i \in S_m \text{ and } m_i(x)\neq C(x)\}|}{|S_m|} \tag{12}$$

其中, x 为输入样本, S_m 为生成的突变模型集合, m_i 为第 i 个突变模型, $m_i(x)$ 为第 i 个突变模型的输出, $C(x)$ 为原始模型输出的标签, $\varsigma(x)$ 可以衡量输入样本 x 对于突变 DNN 模型的敏感性.

基于正常样本以及对抗样本的 LCR 差异, 同时考虑到检测的效率, 他们提出了一种基于标签变化率的判别机制, 该机制参考了统计模型中的检验思想. 使用假设检验来确定两个互斥假设的真实性, 分别为判定该样本为对抗样本的假设以及判定该样本为正常样本的假设:

$$\begin{cases} H_0: & \varsigma(x)>\tau \\ H_1: & \varsigma(x)\leqslant\tau \end{cases} \tag{13}$$

其中, $\varsigma(x)$ 代表样本 x 的 LCR. τ 代表阈值. 采用 3 个标准附加参数 α 、β 以及 δ 用来控制误差的概率, 也就是接受其中一个假设的出错概率. 对于一个样本, 检测算法不断地生成突变模型并且统计其标签变化情况, 直至接受或者拒绝一个假设. 他们使用序贯概率比检验 (sequential probability ratio test, SPRT) 进行测试, 将同一个输入样本 x 输入至突变模型群, 每当一个突变模型的预测标签与原模型的预测标签不同时, 计算并更新 SPRT 概率如下:

$$pr=\frac{p_1^v(1-p_1)^{n-v}}{p_0^v(1-p_0)^{n-v}} \tag{14}$$

其中, $p_1=\tau-\delta$, $p_0=\tau+\delta$, n 代表当前测试过的突变模型数量, v 代表使得输入样本 x 的预测标签不同于原模型的突变模型数量.

最终, 当 pr 满足以下条件时, 结束测试并接受其中一个假设:

$$\begin{cases} pr\leqslant\dfrac{\beta}{1-\alpha} & \text{假设1} \\ pr>\dfrac{1-\beta}{\alpha} & \text{假设2} \end{cases} \tag{15}$$

其中, 假设 1 表示以误差概率为 β 判定输入 x 为对抗样本; 假设 2 表示以误差概率为 α 判定输入 x 为正常样本. 若不满足上述条件则继续使用新的突变模型对测试样本进行分类并不断更新 pr, 直至所有突变模型运行预测完成.

因此, 突变模型测试的具体检测步骤如下.

1) 将待测样本输入原始模型, 获取原始预测标签.

2) 不断将待测样本输入至突变模型, 获取并标签, 实时计算样本的标签变化率.

3) 判断当前标签变化率是否满足 SPRT 假设; 若满足, 则根据假设给出检测结果.

Wang 等人使用 FGSM、JSMA、CW、DeepFool 攻击方法在对他们的方法进行评估. 在 MNIST 数据集上, 使用 GF、NAI、NS 和 WS 算子平均达到了 94.9%、96.4%、83.9% 和 91.4% 检测准确率; 在 CIFAR-10 数据集上使用 GF、NAI、NS 和 WS 算子平均达到了 85.5%、90.6%、56.6% 和 74.8% 检测准确率. 并且突变模型测试在不同类型的攻击算法下检测效果均表现较为稳定. 突变模型测试需要生成大量的突变模型并参与计算, 较为耗费资源, Wang 等人给出的解决方案是预先缓存一组突变模型用于检测, 这样可大幅加快检测速度. 实际上在检测过程中大部分时间耗费在不断地加载突变模型中, 推理时间较长.

3.2.2 降　噪

3.2.2.1 特征压缩

Xu 等人[33]认为深度神经网络的输入本质上有很多“冗余”的特征, 这样会便于攻击者制造对抗样本. 因此, 可以通过比较压缩与未压缩的输入来检测对抗样本. 特征压缩器具有降噪的功能, 正常样本降噪前后模型输出的预测结果基本一致; 而对于对抗样本而言, 降噪前后模型输出的预测结果相差较大. 根据这一特性, 他们提出了特征压缩检测算法, 检测如图 7 所示, 具体检测步骤如下.

1) 将待测样本输入至 DNN 模型, 获得其预测结果 (概率分布).

2) 将待测样本输入至不同的压缩器后降噪后, 再输入至原始 DNN 模型, 获得其预测结果.

3) 比较原始预测分布与不同压缩器降噪后的预测分布之间的 L_1 距离, 取最大值.

4) 比较所得最大距离与阈值的大小, 如果大于阈值, 则判断为对抗样本; 如果小于阈值, 则判断为正常样本.

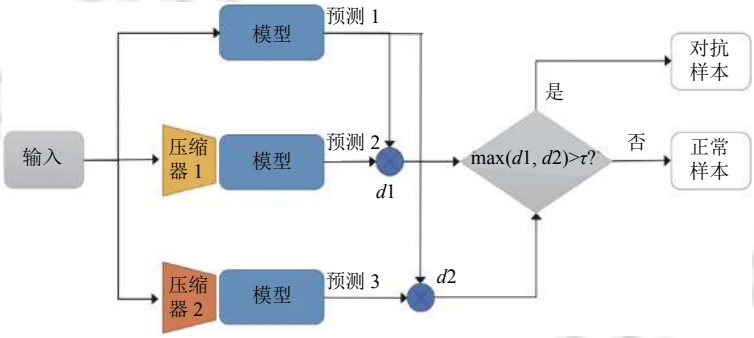


图 7　特征压缩检测框架

特征压缩检测算法的核心是特征压缩器的设计, Xu 等人从减少每个像素的颜色位深度、空间平滑、输入特征降维和有损压缩技术等几个方面进行探索. 并且特征压缩器设计必须满足两个性质: 对于正常样本, 压缩不会显著影响分类器对正常样本的预测结果, 即增加特征压缩器前后正常样本的预测标签不变; 对于对抗样本, 压缩则会显著影响分类器的预测结果, 即增加特征压缩器前后, 对抗样本的预测标签改变.

为了降低色彩深度, Xu 等人使用 $1 \leqslant j \leqslant 8$ 的各种 j 位深度进行了实验. 对于局部平滑, 他们发现中值滤波比高斯和均值滤波更有效; 在非局部平滑中, 他们使用非局部均值去噪[54]. Xu 等人建议使用多种特征压缩方法作为检测模型的一部分, 因为不同类型的压缩器针对组合某些类型的攻击方法可能会更有效. 给定 n 个压缩器 $\{s_1, s_2, \ldots, s_n\}$, 最终输出压缩前后的差异 $d_{\max}$:

$$d_{\max} = \max\left(f(x) - f(s_1(x))_1, \ldots, f(x) - f(s_n(x))_1\right) \tag{16}$$

检测器比较输入与阈值的大小, 如果 $d_{\max} > \tau$ (阈值) 则认为是对抗样本, 否则为正常样本. 其中阈值 τ 根据验证集中假阳率 (正常样本被检测为对抗样本的概率) 不超过 5% 来确定.

Xu 等人使用 FGSM、BIM、JSMA、DeepFool、CW 攻击方法在 3 个数据集上进行了验证: 在 MNIST 数据集上使用减小 1 位色深和大小为 (2×2) 的中值滤波器的组合达到了平均 98.15% 的最好检测准确率, 在 CIFAR-10 数据集使用减小 5 位色深度、(2×2) 的中值滤波器和 (13-3-2) 的非局部均值去噪器组合达到了 84.53% 的最好检测准确率, 在 ImageNet 数据集上使用减小 5 位色深度、(2×2) 的中值滤波器和 (11-3-4) 的非局部均值去噪器组合达到了 85.94% 的最好检测准确率. 但是他们发现, 降低色彩深度对检测 L_2 和 L_∞ 攻击特别有效, 但对 L_0 攻击几乎没有检测效果; 空间平滑方法对检测 L_0 攻击方法较为有效. 此外, 这种检测方法也不擅长于检测扰动较大的对抗样本, 使用上文提到的最佳组合, 在 CIFAR-10 数据集上仅检测到了 18% 和 55% 的 FGSM 和 BIM 对抗样本; 同样的攻击算法在 ImageNet 数据集上为 35.85% 和 55.56%.

特征压缩检测方法的优点在于它的训练成本很低, 而且可以方便地与其他检测或防御手段相结合, 同时 Xu 等人[33]也表明当攻击者了解压缩器的原理时, 他们仍能设计出可以绕开检测的对抗样本.

3.2.2.2 MagNet

Meng 等人[34]提出了 MagNet 防御框架, 除了检测外它还包括改良部分, 也就是传统的防御部分, 在本文中我们主要介绍其检测部分. MagNet 的检测框架如图 8 所示, 具体检测步骤如下.

1) 将待测样本输入至提前训练好的多个降噪自编码器中.

2) 若样本能通过所有降噪自编码器, 则判断为对抗样本; 否则, 判断为正常样本.

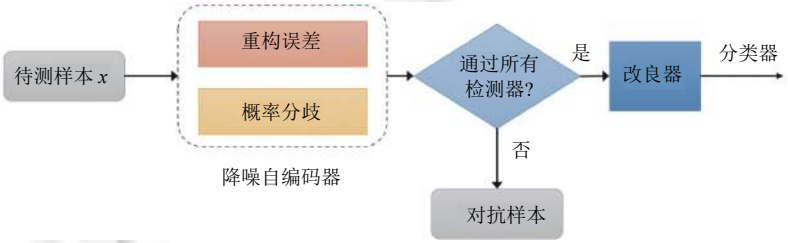


图 8 MagNet 检测框架

Meng 等人设计了两种降噪自编码器它们分别通过重构误差和概率分歧进行训练. 他们认为, 通过训练一个降噪自编码器能够将对抗样本重构为趋于正常的样本并分类正确, 同时其重构后的样本与原始样本相比存在更高的重构误差, 因此能够被检测器识别为对抗样本. 重构误差的定义如下:

$$E(x)=|x-AE(x)|_p \tag{17}$$

其中, $AE(x)$ 表示输入样本 x 经过降噪自编码器的输出, p 代表衡量距离的范数.

首先使用所有的原始样本训练一个自编码器, 使得这个训练集 X_{train} 的损失最小, 其损失函数为:

$$\ell=\frac{1}{|X_{\text{train}}|}\sum_{x\in X_{\text{train}}}x-AE(x)_2 \tag{18}$$

对于一个输入样本, 当重构后样本的重构误差大于阈值时, 则将输入的样本认定为对抗样本. 阈值需要满足如下条件: 当 MagNet 以该阈值对正常样本中的验证集进行检测时, 其假阳率需要低于一个确定的值.

由于某些扰动较小的对抗样本可能在重构后也存在较小的重构误差, Meng 等人认为他们的概率分布应当存在显著的差异, 并对此进行了验证, 提出了基于概率分歧的检测器. 将经过降噪自编码器重构后的样本输入原始的分类器, 定义 $AE(x)$ 为自编码器对于输入 x 的输出. 利用 $f(x)$ 与 $f(AE(x))$ 的分歧程度来判断是否为对抗样本. 其中分歧程度用 Jensen-Shannon 散度来评价. 为了防止某些情况下 $f(x)$ 与 $f(AE(x))$ 的输出可能并没有显著区别, Meng 等人加入了一个温度参数 TE, 将其加入 *Softmax* 层的计算:

$$Softmax(Z(x)_i)=\frac{\exp(Z(x)_i/TE)}{\sum_{j=1}^{n}\exp\left(Z(x)_j/TE\right)} \tag{19}$$

其中, $Z(x)_i$ 表示为 Logits 向量的第 i 维. 当重构误差使用不同的度量方式时, 例如 L_0 距离, L_1 距离以及 L_2 距离, 可

以得到不同的重构误差检测器. 同理, 温度系数TE的不同选择也能够设置多个概率分歧检测器. 因此针对不同复杂程度的任务, 可以设置不同数量及类型的检测器以满足检测效果.

Meng 等人使用了 FGSM、DeepFool 以及 CW 攻击算法在 MNIST 和 CIFAR-10 数据集上验证了 MagNet 框架的防御与检测效果. 对于 MNIST 数据集, Meng 等人设置了两个分别基于L_1距离以及L_2距离的重构误差检测器, 阈值设置为验证集中假阳率不超过 0.1%, 达到了 99.4% 检测精度; 而在 CIFAR-10 数据集上, Meng 等人设置了一个基于重构误差的检测器以及两个温度系数TE分别为 10 以及 40 的概率分歧检测器, 阈值设置为验证集中假阳率不超过 0.5%, 达到 90.6% 检测精度. 由于 MagNet 框架同时存在防御与检测部分, 检测出对抗样本同样视作一种成功的防御. MagNet 在 Meng 等人设置的在所有攻击及不同攻击参数组合下, 均达到了 75% 以上检测精度, 其中有一半以上的攻击达到了 90% 以上的检测精度. 此外, MagNet 框架对于正常样本分类的精度下降很小, 由 90.6% 降低至 86.8%, 同时还能够防御一些灰盒攻击. 但是 Carlini 等人[23]发现 MagNet 容易受到替代模型攻击, 他们利用基于L_2的 CW 攻击在 MagNet 上取得了平均 99% 的攻击成功率.

3.2.2.3 自适应降噪

Liang 等人[35]提出了一种直接检测对抗样本的方法. 这种方法的核心思想是: 将扰动看作是一种噪声, 引入标量量化和平滑空间滤波来降低噪声, 同时利用图像的熵作为度量指标从而对不同类型的图像进行自适应降噪. Liang 等人认为, 对于彩色图像添加很小的扰动就可以攻击成功, 而简单的图像例如手写字则需要较大的扰动, 因此针对不同类型的图像需要采取不同的降噪策略. 自适应降噪的检测框架如图 9 所示, 具体检测步骤如下.

1) 输入待测样本x, 计算其熵值.

2) 根据不同的熵值取不同的量化间隔进行标量量化操作, 并仅对熵值较大的图像进行平滑操作.

3) 得到重构后的样本$RD(x)$.

4) 比较重构后的样本$RD(x)$与原输入样本x在分类器的预测结果, 预测结果不同时, 判断为对抗样本; 否则判断为正常样本.

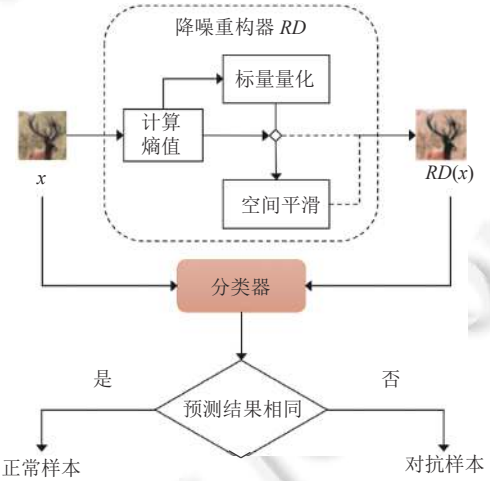


图 9 自适应降噪检测框架[35]

他们在 MNIST 和 ImageNet 两个数据集上对算法的效果进行了评估. 在攻击者未知检测措施的实验中, 他们总共使用了 43 346 个样本进行评估, 其中一半是正常样本, 一半是由 FGSM、DeepFool、CW 攻击生成的对抗样本, 达到了 95% 的平均召回率和 97.81% 的平均检测精度, $F1$-Score 达到了 96.39%, 假阳率为 2.13%.

Liang 等人[35]表明, 他们的方法无需对原始模型进行任何改动, 可以直接与任何现成的模型集成, 作为输入样本的预处理器. 并且他们的方法并不针对特定的攻击, 可以通过熵的大小来自动调整对抗样本的检测策略. 但是这种检测算法的性能与原始模型的分类能力密切相关, 若正常样本在该分类器下本身就具有较低的置信度, 经过降

噪后很容易被误判. 同时, 例如 JSMA 攻击可能会对单个像素值有大幅度的改动, 使用降噪器很难降低该重度扰动的影响.

目前所做的基于神经网络的各种图像分类上的任务, 其能力很大程度上受限于训练数据, 而训练数据都是经过挑选的, 并不能完全反映真实世界的各种场景, 对所有的图像进行统一处理的方式具有一定的局限, 这种根据熵大小对输入进行不同处理的思想一定程度上提供了算法的自适应能力, 增强了算法应对未知输入的处理能力, 值得各种检测算法进行借鉴. 他们的研究也给后续研究带来了思路: 采用经典的图像处理技术或许可以有效地防御对抗样本.

3.2.3 生成式对抗网络

3.2.3.1 Defense-GAN

Samangouei 等人[37]则在 GAN 框架的基础上提出了一种对抗样本检测方法 Denfense-GAN. 其框架如图 10 所示.

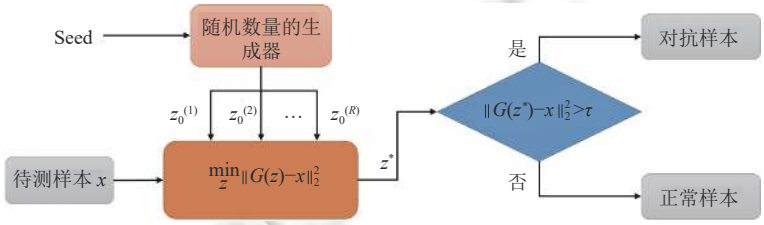


图 10 Defense-GAN 检测框架

他们的 GAN 框架在对抗性环境中同时训练了两个模型: 一个模拟正常样本分布的生成模型, 另一个预测某个输入是正常样本还是对抗样本的检测模型 (即判别器). 生成模型从低维向量 $z \in \mathbb{R}^k$ 到高维输入样本空间 $\mathbb{R}^n$ 学习映射 G. 在 GAN 训练期间, 鼓励 G 生成类似于训练数据的样本. 因此, 可以预期, 符合要求的样本将接近 G 范围内的某个点, 而对抗样本将远离 G 的范围. 给定与训练的 GAN 生成器 G 和要分类的图像 x, 首先找到 z^* 以便将其最小化:

$$z^* = \min_z \| G(z) - x \|_2^2 \tag{20}$$

由于公式 (20) 是一个高度不凸的最小化问题, 通过使用 R 个不同的随机初始化 z 执行固定数量的 N 次梯度下降 (gradient descent, GD) 来近似它. Samangouei 等人认为, 正常样本应当比对抗样本更接近生成器的范围, 因此可以通过比较样本从式 (20) 得到的均方误差作为度量, 来判断样本是否为对抗样本. 具体检测步骤如下.

1) 使用 GAN 框架训练生成模型和判别模型.

2) 将样本输入生成模型, 计算均方误差.

3) 比较均方误差与阈值的大小, 若大于阈值, 判断为对抗样本; 小于阈值, 判断为正常样本.

Samangouei 等人使用 FGSM 和 CW 攻击算法上评估了他们的检测方法, 在 MNIST 数据集上取得了平均 97.38% 的检测准确率, 在 F-MNIST 数据集上取得了平均 82.53% 的检测准确率.

Defense-GAN 可以与绝大部分 DNN 模型结合使用, 并且不修改其本身的分类结构, 可以将其视为分类之前的附加步骤或预处理步骤. 然而, 为了获得更高的精度, Defense-GAN 需要尝试更多的种子输入并运行更多的梯度下降迭代, 这将花费大量时间. 它的性能还取决于 GAN 的质量, 这给训练复杂的任务带来巨大的挑战.

3.2.3.2 GANomaly

Akcay 等人[38]参考了 GAN 的思想, 提出一种称为 GANomaly 的对抗样本检测体系, 它通过学习图像和潜在的空间向量捕捉训练数据的分布.

该检测机制的整体框架如图 11 所示, 包括两个编码器, 一个解码器和一个检测器 (即判别器), 并且定义了 3 种损失函数: 重构误差损失、攻击损失和编码器损失.

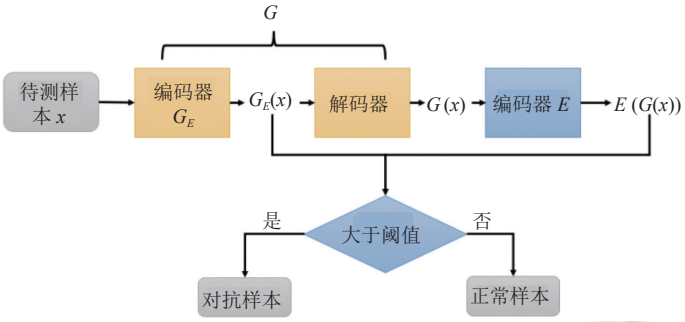


图 11 GANomaly 检测框架

定义重构误差损失如下:

$$\mathcal{L}_{\text{con}} = \mathbb{E}_{x\sim p_X} \| x - G(x) \|_1 \tag{21}$$

其中, $\mathbb{E}$ 代表数学期望, $G(x)$ 代表重构后的样本, 该损失使得编码器解码器对于输入样本得到的重构样本尽可能地与输入样本类似.

攻击损失: 定义对抗损失来计算原始图像以及生成图像的 L_2 距离:

$$\mathcal{L}_{\text{adv}} = \mathbb{E}_{x\sim p_X} \| f(x) - \mathbb{E}_{x\sim p_X} f(G(x)) \|_2 \tag{22}$$

编码器损失:

$$\mathcal{L}_{\text{enc}} = \mathbb{E}_{x\sim p_X} G_E(x) - E(G(x))_2 \tag{23}$$

其中, $G_E(x)$ 代表样本 x 经过解码器的输出, $E(G(x))$ 代表将重构样本 $G(x)$ 输入到解码器 E 后产生的解码样本. 最终的损失函数为以上 3 个损失函数的组合:

$$\mathcal{L} = w_{\text{adv}}\mathcal{L}_{\text{adv}} + w_{\text{con}}\mathcal{L}_{\text{con}} + w_{\text{enc}}\mathcal{L}_{\text{enc}} \tag{24}$$

其中, w_{adv}、 w_{con} 以及 w_{enc} 分别代表各损失项的权重.

GANomaly 的检测步骤如下.

1) 输入待测样本 x 至编码器-解码器-编码器架构, 得到其重构后的样本.

2) 计算样本的异常得分.

3) 比较异常得分与阈值的大小, 大于阈值, 则判断为对抗样本; 小于阈值, 则判断为正常样本.

步骤 2) 中异常得分计算方式如下:

$$\mathcal{A}(x) = G_E(x) - E(G(x))_1 \tag{25}$$

其中, x 代表测试样本. 最终设置阈值, 当异常得分大于该阈值时, 判定该样本为对抗样本.

Akcay 等人在 UBA 数据集上进行了实验, 数据集包括 3 类, 分别为刀、枪以及枪的组件, 检测的 AUC 指标达到了 64.3%, 在 FFOB 数据集上 AUC 指标达到了 88.2%.

Akcay 等人表明, 他们的检测方法能够适应不同复杂度的数据集, 具有较强的泛化能力, 与已有的基于监督学习的检测方法相比无疑是一种提升, 这也使得对抗样本检测任务的应用场景进一步扩大. 但是该方法非常容易受噪声影响, 需要在自编码器上加各种约束, 才能得到一个可用的检测模型.

3.2.4 神经网络特性

3.2.4.1 I-defender

Zheng 等人[39]提出了一种名为 I-defender 的对抗样本检测方法, 该方法探索了 DNN 分类器的内在属性之一, 隐藏状态分布. I-defender 对 DNN 分类器的隐藏状态 (即隐藏神经元的输出) 的分布进行了建模, 隐藏状态空间的维数通常比输入空间的维数低得多, 这使得隐藏状态分布比输入分布更易于建模. 他们认为, 当 DNN 分类器错误地将一个特定的类标签分配给一个对抗样本时, 它的隐藏状态与给定的同一类的正常样本有很大的不同. I-defender 检测框架如图 12 所示.

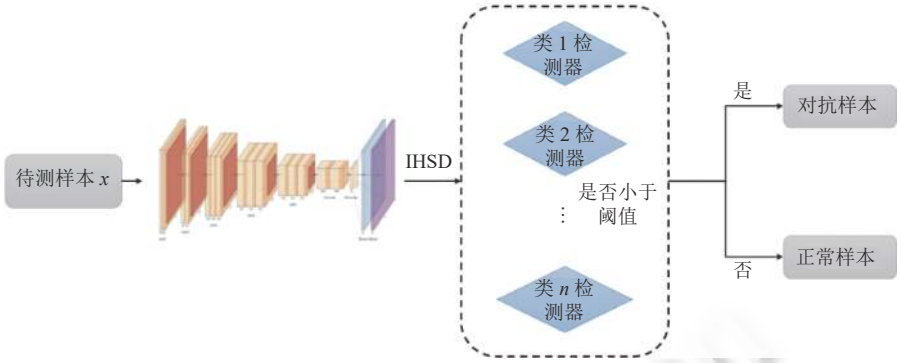


图 12 I-defender 检测框架

他们把正常样本表示的 DNN 的隐藏分布称为内在隐态分布 (intrinsic hidden state distribution, IHSD), 它表征了 DNN 的某些内在性质. 因为 IHSD 无法直接获得, I-defender 使用高斯混合模型 (GMM) 来近似每个类的 IHSD:

$$p(\mathcal{H}(x)\mid\theta,t)=\sum_{k=1}^{K}w_i\mathcal{N}(\mathcal{H}(x)\mid\mu_{tk},\Sigma_{tk}) \tag{26}$$

其中, $\mathcal{H}(x)$ 表示输入 x 在 t 类上的隐藏状态, θ 表示 DNN 的参数, μ_{tk} 和 Σ_{tk} 为第 t 类混合模型中第 k 个高斯分量的均值和协方差矩阵.

在训练完 DNN 分类器后, 将所有训练样本输入其中, 并收集相应的隐藏状态, 以便使用 EM 算法为每个类训练 GMM.

在 I-defender 中, 所有的 DNN 分类器由卷积层和全连接层组成. 卷积层的状态是基于位置的, 这使得直接建模变得非常重要. 因此, 他们选择只对完全连接的隐藏层的状态进行建模. 对于每个类别 t, 分别通过该类正常样本的 IHSD 确定阈值 τ_t. 比较输入样本的 IHSD 是否小于阈值来区分对抗样本.

$$\begin{cases} p(\mathcal{H}(x)\theta,t)<\tau_t, & \text{对抗样本} \\ p(\mathcal{H}(x)\theta,t)>\tau_t, & \text{正常样本} \end{cases} \tag{27}$$

I-defender 的完整检测步骤如下.

1) 将待测样本输入 DNN 模型, 获取样本的预测标签以及 IHSD.

2) 根据样本的预测标签选择对应的检测器.

3) 比较样本的 IHSD 与阈值的大小, 若小于阈值, 判断为对抗样本; 大于阈值, 判断为正常样本.

Zheng 等人使用 FGSM 与 BIM 攻击方法在 CIFAR-10 (使用 34 层的残差网络[55]) 数据集进行白盒攻击评估时对比了 SafetyNet 方法. 结果表明, 虽然部分检测效果不如 SafetyNet, 但 I-defender 在应对不同参数的攻击扰动下表现更加稳定, 达到了 83% 的平均检测精度, 优于 SafetyNet 的 78.7%. 在黑盒攻击实验中, 在 MNIST 数据集上使用 FGSM 攻击算法生成对抗样本进行评估, I-defender 表现出了与第 3.2.3.1 节中的 Defense-GAN 相近的检测效果, 平均检测准确率均达到了 97% 以上, 但运行时间却是 Defense-GAN 的 $1/10^5$, 效率大大提升.

I-defender 这种方法既不需要提前知道攻击方法, 也不需要使用对抗样本训练分类器, 因为它使用的是 DNN 的内部特性, 因此可以跨领域移植 (例如文本领域), 可以很容易地集成到任何基于 DNN 的分类器中, 也可以很容易地与任何现有的防御策略相结合. 但是这种方法需要对每个类别分别确定阈值, 且阈值的确定依赖于训练数据, 无法保证能够找到最优的阈值, 试错成本高.

3.2.4.2 Taboo Trap

Shumailov 等人[40]提出了一种新的对抗样本检测方法, 称为禁忌陷阱 (taboo trap). 他们将一些异常现象称为禁忌 (taboo). 同时根据禁忌条件在训练 DNN 模型的过程中加入一些限制, 当对抗样本输入到模型时, 通常会引起禁忌行为因此被检测. 图 13 为该方法的检测框架.

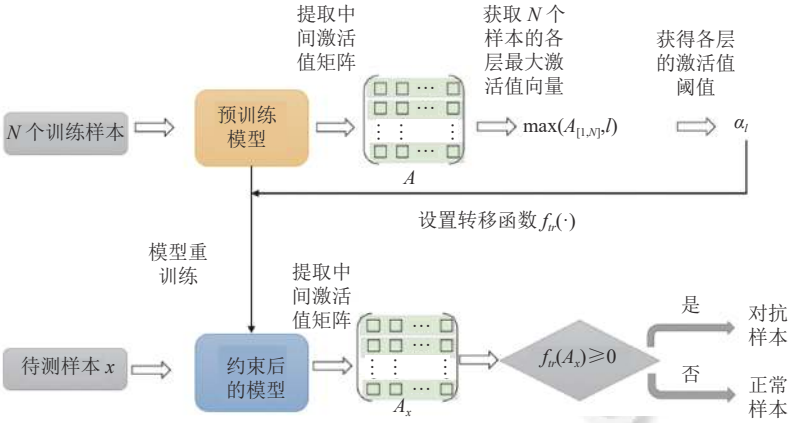


图 13 禁忌陷阱检测框架

要设置一个禁忌陷阱, 首先需要对 DNN 模型各层的激活值进行分析, 并选择一个转换函数接受激活值并参与模型的重训练来限制正常样本的激活值在特定的范围内. 在模型重训练之后的检测阶段, 对于一个未知的输入样本, 如果模型任意一层的激活值超出预期范围, 那么该输入样本被认定为对抗样本.

具体地, 他们定义了一个转移函数 $f_{tr}(A_x)$. 其中 A_x 代表样本 x 在模型中间层的激活值矩阵, 其维度为 $L\times WH\times HE\times CH$, L 为模型层数, WH、HE、CH 分别代表宽和高以及通道数. 在重训练阶段, 该转移函数的输出将被考虑为目标函数的一项并参与训练. 因此对于一批数据 B, 得到以下的重训练损失函数:

$$Loss = L_{\mathrm{SGD}} + \lambda \sum_{x=1}^{B} f_{tr}(A_x) \tag{28}$$

其中, L_{SGD} 为随机梯度下降所引起的损失, λ 是一个超参数. 因此训练后的模型将尽可能使得转移函数的输出为较小的值. 因此, 设置禁忌陷阱的关键在于转移函数的选取, Shumailov 等人利用了第 n 个百分位设置了阈值并构造了转移函数如下:

$$f_{tr}(A_x) = \lambda \sum_{l=0}^{L-1} \sum_{wh=0}^{WH_l-1} \sum_{he=0}^{HE_l-1} \sum_{ch=0}^{CH_l-1} f_p(A_{x,l,wh,he,ch}, \alpha_l) \tag{29}$$

$$f_p(a,b) = \begin{cases} 1, & \text{if } a \geqslant b \\ 0, & \text{else} \end{cases} \tag{30}$$

其中, $A_{x,l,wh,he,ch}$ 代表样本 x 在 l 层的单维的激活值, λ 是一个超参数, wh, he, ch 分别代表宽度, 高度以及通道数的具体维度. α_l 为最大激活值的第 n 百分位阈值, 计算方式如下:

$$\alpha_l = g(\max(A_{[1:N],l})) \tag{31}$$

其中, $A_{[1:N],l}$ 代表所有样本在 l 层激活值的最大值向量, N 为总样本数, g 代表第 n 百分位计算操作. 因此在模型重训练的过程中, 将会惩罚超过阈值的激活值并以此约束模型. 在检测阶段, 对于对抗样本 x', 根据转移函数的输出即可判定检测结果如下:

$$D(x) = \begin{cases} \text{True}, & \text{if } f_{tr}(A_{x'}) \geqslant 0 \\ \text{False}, & \text{else} \end{cases} \tag{32}$$

Taboo trap 的具体检测步骤如下.

1) 输入训练数据, 提取中间层的激活矩阵, 获取最大激活向量.

2) 获得各层的激活值阈值, 并设置转移函数重训练模型.

3) 将待测样本输入至重训练后的约束模型.

4) 提取中间层激活矩阵, 输入至转移函数计算, 若结果大于 0, 则判断为对抗样本; 若结果小于 0, 则判断为正常样本.

他们在 MNIST 以及 CIFAR-10 数据集上针对 FGSM、PGD 以及 DeepFool 攻击算法验证了 taboo trap 的检测效果. 实验结果表明 taboo trap 在 FGSM、PGD 攻击下取得了 94% 的平均检测精度, 而在 DeepFool 下仅取得了 1% 的检测精度, taboo trap 仅能检测较为简单的攻击方法. Shumailov 等人在论文中使用较为简单的转移函数, 这可能是 taboo trap 无法检测由 DeepFool 生成的对抗样本的原因, 这也表明 taboo trap 比较依赖于转移函数的设置, 泛化能力较差.

总体来说禁忌陷阱是一种较为新颖的对抗样本检测方法, 虽然原始的禁忌陷阱在检测效果上面对 DeepFool 等较强的攻击方法表现较差, 然而由于其无需引入额外的参数, 唯一的额外开销来自训练时间的增加, 在推理速度上优于 MagNet[34]等检测框架. 同时禁忌陷阱可以设置不同的禁忌形成联合检测, 这大大增加了攻击者针对检测机制的攻击难度.

3.2.5 特征对齐

3.2.5.1 UnMask

Ilyas 等人[56]和 Tsipras 等人[57]认为深度学习模型的脆弱性是由于模型对数据中一般特征过于敏感导致的. 通常在训练模型时为了最大程度地提高准确性, 分类器会使用任何可用的信息 (其中包括了很多人类无法理解的特征). 这些人类难以理解的 (非鲁棒) 的特征虽然有助于提高准确性, 但很容易被利用来设计对抗样本. Freitas 等人[41]扩展了上述概念, 他们认为分类模型的脆弱性是由非鲁棒特征导致的, 并且提出了一种基于鲁棒特征对齐的对抗样本检测框架 UnMask.

UnMask 通过从图像样本 (例如“鸟”) 中提取鲁棒性特征 (例如喙, 翅膀, 眼睛) 并将其与分类的预期特征进行比较来检测对抗样本. 例如, 一张“自行车”的图像提取出来的鲁棒性特征可能是车轮、踏板车架, 然而这个样本被 DNN 分类为鸟类, 显然这些鲁棒性特征与鸟类的鲁棒性特征不存在交集, 则这个样本很有可能是对抗样本. Freitas 等人构建了一个新的数据集 UnMaskDataSet, 由 4 个部分组成: PASCAL-Part、PASCAL VOC 2010、ImageNet 和 Flickr, 并且对其定义了 3 种特征: 有用的特征、鲁棒性特征、有用但非鲁棒性特征, 使用 Mask R-CNN[58]作为特征提取模型 K. 例如, 对于车轮特征, 训练数据将包括自行车图像和指示该图像的哪个区域表示车轮的分割掩码.

如图 14 所示为 UnMask 算法的检测框架, 其检测步骤如下.

1) 训练一个鲁棒特征提取器 K.

2) 给定输入 x, 用特征提取器 K 提取其鲁棒性特征 $f_r = K(x)$, 同时分类器对 x 分类得到类别 t.

3) 计算提取出的鲁棒性特征 f_r 和类别 t 应该有的鲁棒性特征集 f_t 之间的 Jaccard 相似度评分 $score = JS(f_t, f_r)$, 如果 $1 - score$ 大于阈值 τ 则认为输入 x 为正常样本, 否则为对抗样本.

4) 如果检测判断为对抗样本, UnMask 通过比较 f_r 与所有类别鲁棒性特征集之间的相似度评分 $score$, 取分数最高的类别作为分类结果.

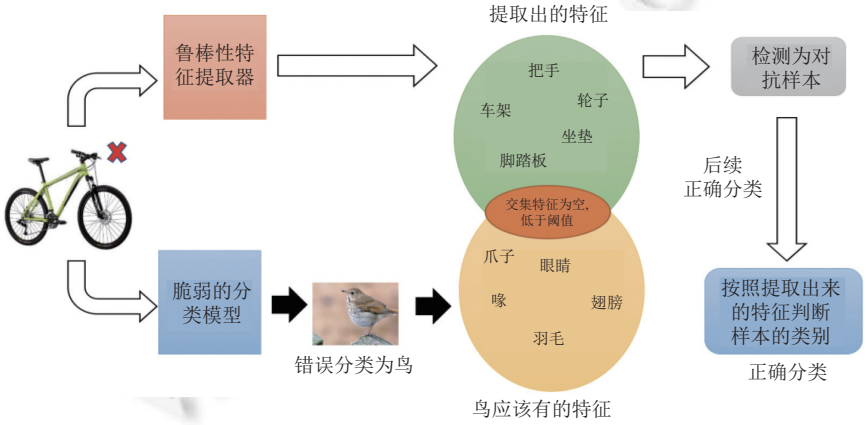


图 14 UnMask 检测框架

Freitas 等人使用 FGSM 和 PGD 攻击算法 (改变攻击参数) 在 ResNet 和 DenseNet 两个模型上评估了 UnMask, 分别取得了 82.5% 和 80.4% 的检测准确率, 其中 UnMask 对扰动参数大的 L_∞ 攻击生成的对抗样本检测效果较差, 平均检测精度为 60.6%.

UnMask 算法为我们提供了一种新颖的研究方向, 它成功地将鲁棒性特征和人类的认知直觉联系起来, 提供了一种强大的、可解释的检测方法. 这种方法检测迅速, 并且与模型的架构无关, 易于集成. 但是 UnMask 也有一些明显的缺陷: 需要人为标注规定鲁棒性特征构造数据集, 需要一定的人工成本; 仅使用了基于梯度的对抗攻击方法进行评估, 不够全面; 判断是否为对抗样本的阈值需要寻优确定.

3.3 有监督检测

3.3.1 统计方法

3.3.1.1 Logits 重构

在第 3.2.1.2 节中 Hendrycks 等人[30]已经证明了 *Softmax* 预测概率可用于检测对抗样本, 他们在此基础上利用有监督的方式设计检测器, 提出了基于 Logits 重构的检测算法, 该方法的检测框架如图 14 所示. 他们在分类器模型中添加辅助解码器[59], 以 Logits 作为输入重构图像, 得到重构后的输入. 解码器和分类器只在正常样本上进行联合训练. 检测可以通过创建一个检测器网络来完成, 该网络以重构输入、Logits 和置信度为输入, 并输出输入样本是对抗样本的概率, 该检测器网络使用正常样本和对抗样本进行二分类训练. 与正常样本相比, 对抗样本的重构输入拥有更大的噪声, 因而可以作为检测对抗样本的一种方式.

Logits 重构检测算法的检测框架如图 15 所示, 具体步骤如下.

1) 将待测样本输入 DNN 模型, 获取其 Logits.

2) 将 Logits 输入至辅助解码器, 得到重构后的输入.

3) 将原始 Logits、置信度和重构后的输入共同输入至检测器, 检测器输出结果.

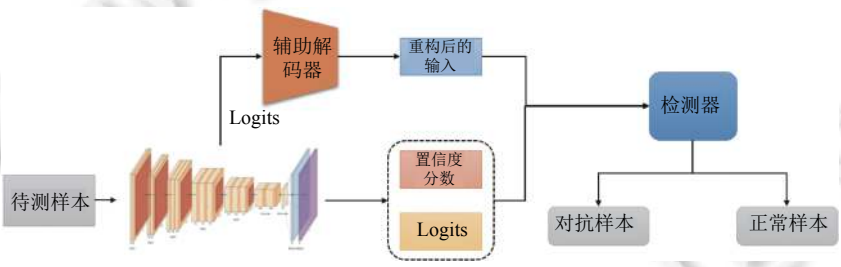


图 15 Logits 重构输入检测框架

Hendrycks 等人[30]在 MNIST 数据集上评估了这种方法, 但并没有直接给出实验结果, 只是表明可以检测 FGSM 和 BIM 生成的对抗样本. 但是在白盒攻击以及了解检测器的情况下, 攻击者仍然可以找到同时欺骗分类器和检测器网络的对抗样本. Logits 重构输入与第 3.2.1.1 节中的 PCA 主成分分析和第 3.2.1.2 节中的 *Softmax* 分布方法是 Hendrycks 等人在对抗样本检测领域早期提出的检测方法, 虽然它们检测效果并不出色, 泛化能力也较弱, 但他们方法给后续对抗样本检测领域的研究提供了方向, 具有十分重大的引领意义.

3.3.1.2 核密度估计

Feinman 等人[42]认为对抗样本与正常样本的分布之间存在一定的距离 (对抗样本存在于稍远离正常样本的流形区域之外). 许多训练数据实际存在于高维空间的低维流形区域. 对抗扰动并没有改变原始数据的真实标签 (潜在标签), 它只是将数据推移出了正常样本的数据流形.

基于上述理论, Feinman 等人[42]提出了一种对抗样本的检测方法: 核密度估计 (kernel density estimate, KDE). 他们使用高斯混合模型对深度神经网络隐藏层最后一层的输出进行建模. 具体地, 给定输入样本 x 分类为标签 t, 那么 x 的核密度估计为:

$$KDE(x)=\frac{1}{|X_t|}\sum_{X_{\text{train}}\in X_t}\exp\left(\frac{|f(x)-f(X_{\text{train}})|^2}{\sigma^2}\right) \tag{33}$$

其中, X_t 是带有标签 t 的训练数据集合, X_{train} 表示训练集样本, σ 为高斯混合模型的内核带宽.

Feinman 等人受到 Bengio 等人[60]和 Gardner 等人[61]工作的启发, 使用高斯内核作为内核函数, 并且在逻辑空间中求值. 如果 x' 远离目标类别流形, 则对抗样本 x' 的 *KDE* 值较低, 因此也可以使用基于阈值的方法进行检测. 选定一个阈值 τ, τ 通过正常样本的 *KDE* 值确定, 如果 $KDE(x)<\tau$, 则认为该样本为对抗样本; 否则, 为正常样本.

核密度估计算法检测框架如图 16 所示, 获取样本 *KDE* 值的具体步骤如下.

1) 输入训练集样本, 获取其 Logits, 并根据样本的预测标签分类.

2) 对每一类的样本分别构建核密度估计模型.

3) 输入待测样本, 获取其预测标签和 Logits.

4) 根据预测标签将 Logits 输入至对应类别的和密度估计模型, 计算 *KDE*.

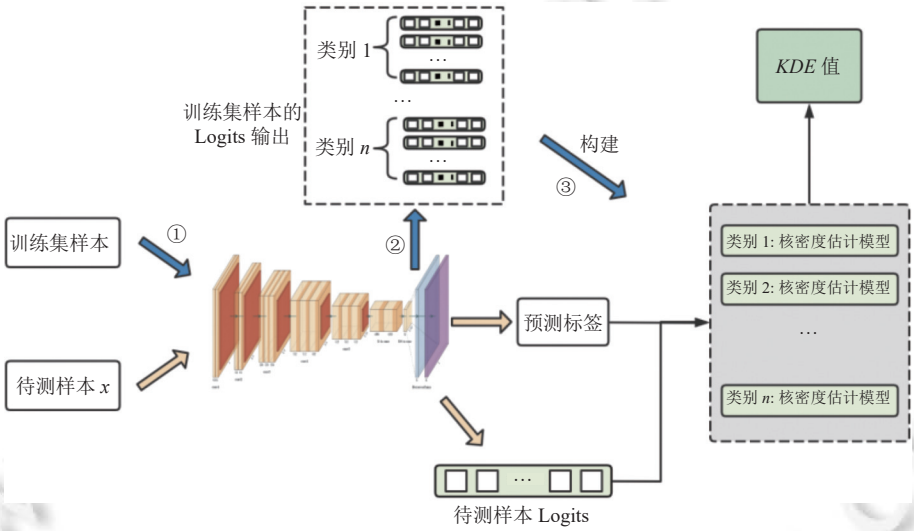


图 16 KDE 检测框架

核密度估计检测算法是和贝叶斯不确定性估计算法一同提出的, 作者最终采用了联合两个指标训练检测器的形式, 具体检测效果将在第 3.3.1.2 节中介绍. Carlini 等人[19]的工作表明, *KDE* 阈值检测在 CIFAR-10 数据集上检测效果较差, 可以通过在攻击时添加约束项进行规避.

3.3.1.3 贝叶斯不确定性估计

虽然核密度估计可以有效检测远离类流形的样本, 但是当对抗样本接近目标类流形时, 这种方法就可能会失效. 因此, Feinman 等人[42]提出了第 2 种对抗样本检测方法: 贝叶斯不确定性估计 (Bayesian uncertainty estimate, BUE). 这种方法可以测量神经网络对给定输入的不确定性. 他们不依赖于神经网络反馈的置信度, 而是向网络中添加随机性. 他们认为正常样本在随机选择的情况下依旧能被分类为相同的类别, 而对抗样本并不总是能够被分为相同类别. Feinma 等人[42]参考了 Gal 等人[62]的工作, 在模型推理阶段使用 Dropout[63]作为衡量不确定性的方法. 他们重复将样本输入至随机网络 f_{random} (启用了 Dropout 功能) N 次, 从而量化网络的不确定性:

$$BUE(x)=\left(\frac{1}{N}\sum_{i=1}^{N}(\|f_{\text{random}}(x)\|_2)^2\right)-\left(\left\|\frac{1}{N}\sum_{i=1}^{N}f_{\text{random}}(x)\right\|_2\right)^2 \tag{34}$$

贝叶斯不确定性估计也就是计算对随机网络的 N 个输出的每个分量的方差之和, 因此如果每次预测 $f_{\text{random}}(x)$ 的结果都相同, 那么 *BUE* 为 0. 这种方法通过选择一个阈值 τ 来进行检测, 其中 τ 根据正常样本的 *BUE* 值确定, 如

果 $BUE(x)>\tau$ 则认为该样本为对抗样本, 否则为正常样本. 并且该算法对 N 的选取并不敏感, 通常只需要取 $N>20$ 即可.

贝叶斯不确定性估计检测框架如图 17 所示, 获取 *BUE* 的具体步骤如下.

1) 输入待测样本至开启了 Dropout 的原始模型.

2) 获取待测样本 n 次输出的概率分布.

3) 按列求方差, 按行求均值, 得到 *BUE* 值.

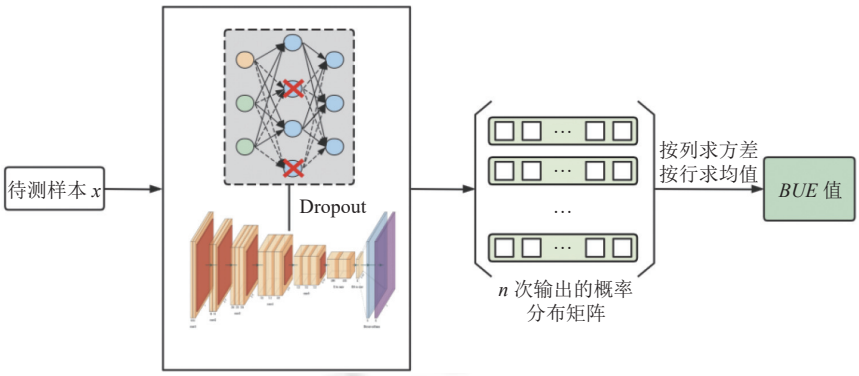


图 17 *BUE* 检测框架

Feinman 等人训练了一个简单的逻辑回归分类器, 具有两个输入特征: *KDE* 和 *BUE*, 并使用 FGSM、BIM、JSMA、CW 这 4 种攻击方法在 3 个数据集上进行了评估: 在 MNIST (使用 LeNet 模型[64]) 数据集下的 AUC 指标达到了 92.59%; 在 SVHN (使用 LeNet 模型附加额外的中间全连接层) 数据集下的 AUC 指标达到了 85.54%; 在 CIFAR-10 (12 层深度卷积神经网络) 数据集下的 AUC 指标达到了 90.20%, 表明这种组合方法可以有效检测多种数据集下不同类型的对抗样本.

Feinman 等人[42]的方法在组合 *KDE* 与 *BUE* 的情况下具有更加全面的检测效果. 同时 Carlini 等人[19]的工作指出使用 CW 击生成能够欺骗 *BUE* 的对抗样本所需要的扰动更大, 因此与其他方法相比 *BUE* 更难被攻破, 并且作为模型的附加结构部署相对简单.

3.3.1.4 局部内在维度

通过维度扩展模型评估数据的局部维度结构已经成功应用于许多场景, 如流形学习、维度缩减、相似性搜索和异常检测[65,66]. 早期的扩展模型将内在维度描述为数据集的一个特性, 但 Houle 等人[66]将局部内在维度 (local intrinsic dimensionality, *LID*) 这一概念推广到从参考点到其邻域的局部距离分布, 通过累积分布函数的增长特性揭示了参考点附近局部数据子流形的维数. Ma 等人[43]受到他们工作的启发, 使用 *LID* 来描述对抗样本所处区域的内在维度, 并使用 *LID* 的估计值检测对抗样本.

由于真实世界的数据集并不是均匀分布的, 数据流模型不能完美的适用, Ma 等人通过计算输入样本 x 到所有样本中 k 个最近邻居的距离估算样本的 *LID* 值. 他们使用了 Amsaleg 等人[67]开发的最大似然估计器 (maximum likelihood estimator, MLE), 因为它是统计效率和复杂性之间的折中方案. 给定参考样本 $x\sim P$, 其中 P 代表数据分布, x 处的 *LID* 估计值如下:

$$LID(x)=-\left(\frac{1}{k}\sum_{i=1}^{k}\log\left(\frac{r_i(x)}{r_m(x)}\right)\right)^{-1} \tag{35}$$

其中, $r_i(x)$ 表示从分布 P 得出的点样本中 x 与第 i 个最近邻居之间的距离, $r_m(x)$ 表示邻居距离的最大值.

LID 检测方法的框架如图 18 所示, 具体步骤如下.

1) 首先对输入的训练集样本进行攻击和添加随机噪声, 生成 3 类样本, 将正常样本、对抗样本和噪声样本输入至 DNN 模型.

2) 使用 MLE 估计器计算所有样本在所有转换层 (包括 Conv2d, Max-Pooling, Dropout, ReLU 和 *Softmax*) 中的 *LID* 值.

3) 将随机噪声样本和正常样本归为一类 (排除随机扰动对检测的影响)、对抗样本一类, 利用这两类样本训练一个二分类的逻辑回归分类器.

4) 输入待测样本, 计算待测样本的 *LID* 特征值, 输入至检测器得到检测结果.

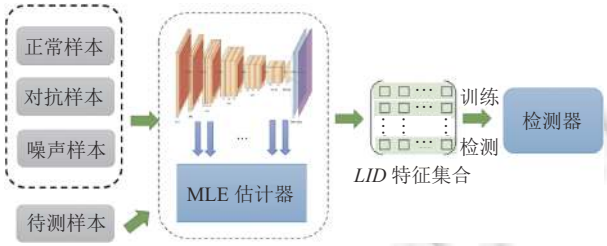


图 18 *LID* 检测框架

步骤 2) 中 MLE 估计器以转换层中神经元的激活值作为输入, 每个样本有 L 个特征值 (每个转换层一个特征), 假设总共输入了 N 个样本, 模型转换层的数量为 L, 那么通过 MLE 估计器得到的 *LID* 特征矩阵维度应该为 $(N,1,L)$.

Ma 等人参照 Feinman 等人[42]的实验条件, 对 *LID* 进行了评估, *LID* 在 MNIST、CIFAR-10、SVHN 数据集上分别取得了 99.24%、98.94% 和 97.60% 的平均 ROC-AUC 得分, 均优于组合下的 *KDE* 与 *BUE*. Ma 等人也表明 *LID* 是表征对抗样本分布非常有潜力的方法.

Lu 等人[68]使用 *LID* 做了两组补充实验, 他们发现: *LID* 的性能对生成对抗样本时的置信度参数十分敏感, 使用不同置信度对抗样本训练的 *LID* 检测器性能会有波动; *LID* 在表征使用另一个 DNN 模型生成的对抗样本的内在维度时能力有限, 在检测该类对抗样本时效果较差.

3.3.1.5 最近邻影响函数

Cohen 等人[44]提出了一种最近邻影响函数的对抗样本检测方法 (nearest neighbor influence functions, NNIF), 该方法适用于绝大部分预先训练好的神经网络分类器, 其检测框架如图 19 所示.

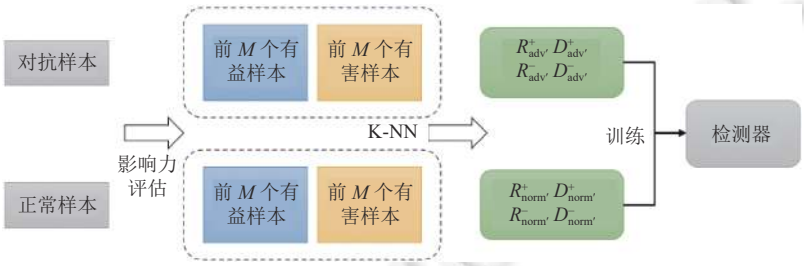


图 19 最近邻影响函数检测框架

最近邻影响函数检测算法利用了神经网络的预测效果受到神经网络隐藏层中训练数据最近邻表现的影响这一特性. 神经网络中的训练数据与神经网络的预测结果存在一定关联. 对预测结果起到积极作用的数据, 我们称其为有益样本, 不利于神经网络预测的训练数据, 我们称其为有害样本. 对于一个正常的输入样本来说, 其 K 最近邻 (K-nearest neighbor, KNN) 的训练样本应该和对其起到有益作用的训练样本存在关联. 相反, 对于对抗样本, 这样的关联性可能会被破坏. 使用了 Koh 等人[69]影响力评估公式衡量一个样本对输入样本的影响. 通过观察样本的有益样本以及 KNN 样本, 他们发现正常样本的有益样本以及 KNN 样本通常高度相关, 其 PCA 特征压缩后通常聚集在一起. 与此相反, 对抗样本的有益样本以及 KNN 样本距离甚远.

基于以上的观察, Cohen 等人[44]提出了最近邻影响函数检测方法: 利用正常样本以及对抗样本的 NNIF 特征训练一个对抗样本检测器, 对于未知的样本, 提取相应的 NNIF 特征并进行对抗样本检测. 对于一个输入样本 x_i: R^+ 代表神经网络的激活层中对于 x_i 最有益的前 M 个样本的最近邻排名; D^+ 代表对于 x_i 最有益的前 M 个样本的 L_2 距离; R^- 代表对于 x_i 最有害的前 M 个样本的最近邻排名; D^- 代表对于 x_i 最有害的前 M 个样本的 L_2 距离. 正常样本的 *NNIF* 特征定义为:

$$NNIF_{\text{pos}} = (R^+_{\text{norm}}, D^+_{\text{norm}}, R^-_{\text{norm}}, D^-_{\text{norm}}) \tag{36}$$

对抗样本的 *NNIF* 特征定义为:

$$NNIF_{\text{neg}} = (R^+_{\text{adv}}, D^+_{\text{adv}}, R^-_{\text{adv}}, D^-_{\text{adv}}) \tag{37}$$

最近邻影响函数检测算法具体步骤如下.

1) 输入包含正常样本与对抗样本的训练集至原始模型.

2) 利用对训练集样本进行影响力评估, 得到 *NNIF* 特征.

3) 将 *NNIF* 特征输入至检测器进行训练.

4) 待测样本输入至原始模型, 计算 *NNIF* 特征.

5) 输入 *NNIF* 特征值检测器得出检测结果.

Cohen 等人在 ResNet-43[70]模型上基于 CIFAR-10 以及 SVHN 数据集进行了 FGSM、PGD、DeepFool、JSMA、CW 攻击算法生成的对抗样本检测实验, 在 CIFAR-10 数据集上的 AUC 均值达到了 99.32%; 在 CIFAR-100 数据集上 AUC 均值达到了 93.55%; 在 SVHN 数据集上的 AUC 值均在 98.92%, 优于同等条件下的局部内在维度检测算法.

然而, NNIF 的不足之处在于其计算量较大. 对于每一个样本都需要评估其余样本对其的影响力以寻找有益的样本以及有害的样本, 因此训练 *NNIF* 检测器将耗费较多时间.

3.3.1.6 高斯过程回归

Lee 等人[45]提出了基于高斯过程回归的检测框架. 他们认为有效的检测方法应该具备以下特征: 首先检测方法需要具备高效的检测效率, 即对抗样本检测算法应该在即使仅有少量训练数据的情况下也能够进行有效检测; 此外, 对抗样本检测方法应该是不可微的, 从而使得基于梯度的攻击方法无法对检测器进行二次攻击. 高斯过程回归检测算法的框架如图 20 所示.

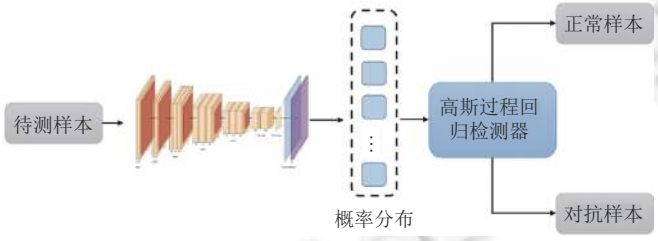


图 20 高斯过程回归检测框架

高斯过程是一个随机过程, 其中每一个有限的随机变量集合都存在一个多元的正态分布, 而高斯过程回归是一种基于当前的观测数据来推断所有数据范围内均值和方差的方法[71−73]. 高斯过程回归考虑到数据间的协方差, 定义了先验概率并预测后验概率. 因此对于遵循高斯过程的数据, 利用高斯回归过程能够在使用少量训练样本情况下实现高效检测.

首先针对正常的图像样本以及对抗样本, 提取预训练好的分类模型生成的特征, 中间特征定义为模型输出的概率分布, 其维度等于类别数. 中间特征集合定义为:

$$X_{\text{inter}} = \{Z(x) \mid x \in \{x_i\} \cup \{x'_i\}\} \tag{38}$$

其中, $\{x_i\}$ 以及 $\{x'_i\}$ 分别代表正常样本的集合和对抗样本的集合.

然后, 基于提取出的特征集合作为高斯过程回归检测器的观测数据, 拟合观测数据如下:

$$\begin{cases} D_{\text{observed}} = \left\{\left(x_{\text{inter}}^{i}, y^{i}\right)\right\} \\ \text{s.t.} x_{\text{inter}}^{i} \in X_{\text{inter}} \\ y^{i} = \begin{cases} 0, & \text{if } x_{\text{inter}}^{i} \text{是正常样本} \\ 1, & \text{if } x_{\text{inter}}^{i} \text{是对抗样本} \end{cases} \end{cases} \tag{39}$$

高斯过程回归检测算法的具体检测步骤如下.

1) 输入待测样本至 DNN 模型, 获取其概率分布.

2) 将概率分布作为特征输入高斯过程回归检测器.

3) 高斯过程回归模型拟合观测数据, 输出检测结果.

Lee 等人在 MNIST (5 层的卷积神经网络) 以及 CIFAR-10 数据集上进行了对抗样本检测的实验. 选取的协方差函数为指数协方差函数[71]. 针对 JSMA、DeepFool 以及 CW 攻击算法, 高斯过程回归检测在 MNIST 数据集上达到 99.1% 的平均检测准确率, 在 CIFAR-10 数据集上达到了 96.9% 平均检测准确率. 针对 BIM 和 FGSM 攻击算法检测效果较差, 平均检测准确率分别为 81.1% 和 63.67%.

他们提出的方法能够基于少量对抗样本进行高效检测, 同时检测方法针对输入样本不可微. 然而目前也存在许多针对不可微防御的攻击方法, 因此该检测方法仍然有被二次攻击的风险. 同时在高斯过程回归中, 相似数据的影响将定义为协方差. 如果数据维度较高, 很难选取协方差的模式. 对此 Lee 等人给出了如下解决方案: 将输入经过卷积以及池化提取低维的特征来代替原始的高维图像, 以此来有效地进行高斯过程回归.

3.3.2 对抗训练

3.3.2.1 *N*+1 类对抗训练

对抗训练是一种防御方法, 它通过在每次迭代训练中将对抗样本注入到训练集中来重新训练模型, 以加强模型的鲁棒性.

Grosse 等人[46]提出了一种对抗训练的变体, 他们没有尝试对对抗样本进行正确分类 (通过将对抗样本添加到训练集并带有正确的标签), 而是引入了一个新的 *N*+1 类 (该类别中只有对抗样本), 并训练分类网络来检测对抗样本例如原来是 0–9 的十分类问题, 令对抗样本的标签为 10, 作为第 11 类训练, 当输入样本的预测标签为 10 时, 表示该样本是一个对抗样本. *N*+1 类对抗训练的检测框架如图 21 所示, 具体步骤如下.

1) 使用原始训练集 X 训练一个原始分类网络 $f(x)$.

2) 在原始分类网络 $f(x)$ 上, 对每一个 $(x_i, y_i) \in X$ 生成对抗样本 x'.

3) 将对抗样本加入到原始训练集构建新数据集, 使得 $X_{\text{new}} = X \cup \left\{\left(x_i', N+1\right) : i \in |X|\right\}$, 其中 $N+1$ 就是所有对抗样本的标签.

4) 用新数据集 X_{new} 训练一个新分类网络 $f_{\text{new}}(x)$, 用于检测对抗样本及正常样本分类.

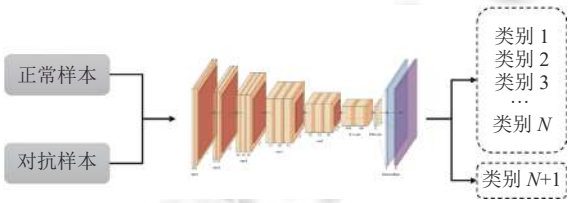


图 21 *N*+1 类对抗训练

他们使用 FGSM 和 JSMA 攻击方法在 MNIST (2 个 5×5 的卷积层的卷积神经网络) 数据集上进行了验证, 在原始模型精度下降不超过 1% 的情况下可达到 99% 左右的检测准确率, 但是 Carlini 等人[19]发现这种方法并不适用于更复杂的数据集, 例如在 CIFAR-10 数据集中假阳率过高.

这种方法将不同类别的对抗样本全都视作一致的对抗样本, 试图利用神经网络捕获对抗样本的某种共性, 但是这种方法没有考虑到不同对抗攻击算法的特性, 例如 FGSM 攻击产生的对抗样本所包含的扰动是原多于

JSMA 攻击的, 使用多种类型的对抗样本训练可能会导致模型难以拟合, 影响原始模型的精度. 并且他们忽略的样本本身的特性, 第 N+1 类训练时包含了太多冗余的信息, 复杂度较高.

3.3.2.2 二元分类器

Gong 等人[47]提出了一种与上述方法类似的检测技术, 不同的是他们构建了一个二元分类检测器 D, 而不是完全重新训练原始分类网络. 这里的二元分类检测器网络与原始网络是完全独立的, 他们使用的对抗样本是通过对原始分类网络攻击生成的, 而不是针对二元分类检测器 D 生成的.

二元分类器的检测框架如图 22 所示, 具体检测步骤如下.

1) 使用正常样本与对抗样本构建为二分类训练集 $X_{\text{new}}=\{(x_i,1):i\in|X|\}\cup\left\{\left(x_i',0\right):i\in|X|\right\}$.

2) 使用二分类训练集训练二分类检测器网络.

3) 输入待测样本至二分类检测器网络, 输出检测结果.

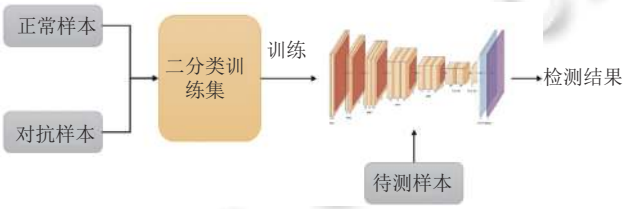


图 22 N+1 类对抗训练

Gong 等人在 MNIST、CIFAR-10 和 SVHN 数据集评估了二元分类器, 对 FGSM 攻击算法生成的对抗样本的平均检测准确率达到了 99%, 但是在进行迁移性评估时发现该方法的检测器网络对使用 FGSM 和 BIM 攻击方法生成的对抗样本的扰动系数 λ 敏感. 基于扰动系数 λ_1 生成的对抗样本训练的检测器无法检测基于扰动系数 λ_2 生成的对抗样本, 尤其是当 $\lambda_2<\lambda_1$ 时. 因此, 利用某一种特定攻击算法生成的对抗样本训练得到的检测器, 无法保证能够对其他攻击方法生成的对抗样本有较好的检测效果. 但是, 在 CIFAR-10 数据集上测试时, 该方法同样存在假阳率过高的问题, 并且容易受到 CW 攻击. Gong 等人也表明二元分类器的优势在于它可以作为任意模型的预处理步骤, 无需对原始模型进行改动或获取其信息, 是一种简单有效的检测方法. 这种方法直接将所有对抗样本统一作为一类训练检测器, 与 N+1 类对抗训练一样, 会有大量的冗余信息干扰检测器的训练.

3.3.2.3 GAT

Yin 等人[48]提出了一种能够自适应的检测机制, 即生成式对抗训练 (generative adversarial training, GAT), 其思想是根据分类器的输出将输入空间划分为若干个子空间, 并且在这些子空间中执行对抗样本检测的任务, 其检测框架如图 23 所示. GAT 首先将正常样本和对抗样本输入到分类器进行分类, 然后将每个类中的对抗样本与正常样本输入到检测器中, 二分类输出检测结果, 对每一个类别 t 训练一个检测器, 其中检测器的训练目标函数如下:

$$\rho(\theta_t)=\mathbb{E}_{x\sim\eta\setminus t}\left[\max_{\delta\in\epsilon}\ell(D_t(x+\delta;\theta_t),0)\right]+\mathbb{E}_{x\sim\eta_t}\left[\ell(D_t(x;\theta_t),1)\right] \tag{40}$$

其中, θ_t 代表 t 类别检测器的参数, D_t 代表 t 类别的检测器, δ 代表扰动量, $\eta\setminus t$ 代表集合 $\{x_i:y_i\neq t\}$, η_t 代表集合 $\{x_i:y_i\neq t\}$, y_i 代表输入样本 x_i 的分类标签, ϵ 代表扰动量的范围, ℓ 代表损失函数, $\mathbb{E}$ 代表数学期望. 内部最大化问题通过迭代 PGD 攻击来实现.

结束训练后, 我们可以获得输入样本以及其对应类的一个 Gibbs 分布:

$$p(x,t)=\frac{\exp(z(D_t(x)))}{\sum_t\int\exp(-E_{\theta_t}(x))dx} \tag{41}$$

其中, $z(D_t(x))$ 代表第 t 类的检测器输出的概率. 最终根据 $p(x,t)$ 设置阈值并且拒绝低概率输入从而完成对抗样本检测. Yin 等人在实验中对于每一个类都使用了一个同样的阈值进行评估, 在实际使用时应该通过优化函数来确定. 阈值优化函数由检测器在验证集的真阳率和假阳率共同约束.

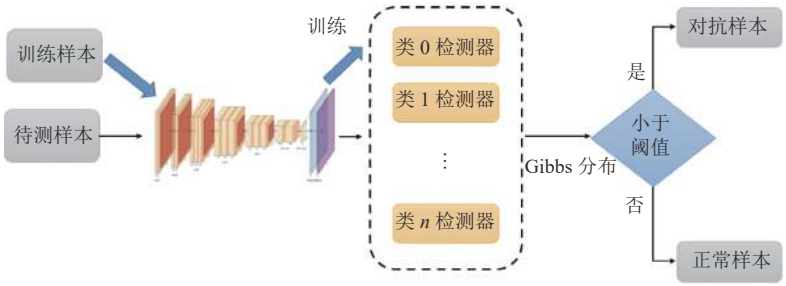


图 23 N+1 类对抗训练

GAT 的具体检测步骤如下.

1) 输入包含正常样本和对抗样本的训练集样本至原始模型, 获取其预测标签.

2) 根据预测标签分类训练二分类检测器.

3) 输入待测样本至原始分类模型获得预测标签.

4) 根据预测标签将样本输入对应的检测器, 获得样本的预测为对抗样本的概率.

5) 根据检测器输出概率计算 Gibbs 分布, 与阈值进行比较, 若小于阈值, 则判断为对抗样本; 若大于阈值, 则判断为正常样本.

Yin 等人在 MNIST 和 CIFAR-10 数据集上针对 PGD 攻击 (使用多组攻击参数) 验证了他们检测方法的有效性, 其中在 MNIST 数据集上, 达到了 99% 的平均检测准确率; 在 CIFAR-10 数据集上达到了 93% 的平均检测准确率. 但是该对抗样本检测方法仅在 PGD 攻击方法上进行测试, 在更多攻击方法下的表现有待评估. 同时, 该方法对于每一个类别都需要独立训练检测器, 训练成本较大, 复杂度高.

3.3.2.4 动态对抗训练

Metzen 等人[49]提供了一个新的思路, 他们没有针对原始样本本身进行检测, 而是通过检测分类网络内部的特征来检测对抗样本. 他们训练了一个检测神经网络, 该检测神经网络从原始分类神经网络的某一层获取输出作为检测器的输入, 并输出样本是否为对抗样本的概率. 他们利用原始分类网络生成对抗样本, 构建由对抗样本与正常样本组成的新数据集 X_{new}, 固定原始分类网络的权重, 获取其中间层的输入并采用交叉熵作为损失函数训练该检测神经网络.

图 24 为其检测框架示意图, 图的上部分为原始分类网络, 延伸出来的检测 (i) 连接到下部分检测网络作为其输入. 具体检测步骤如下.

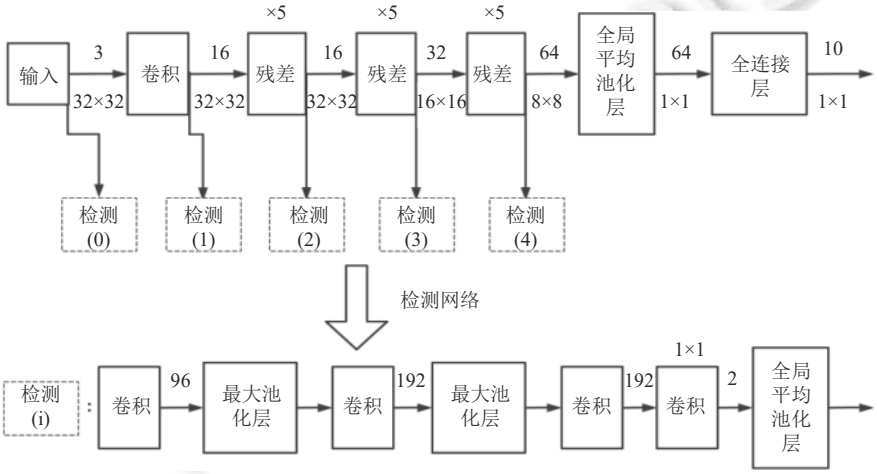


图 24 动态对抗训练检测框架

1) 输入包含正常样本与对抗样本的训练集数据至原始分类网络.

2) 获训练集样本的中间层输出, 将输出输入至检测网络进行训练.

3) 输入待测样本至原始分类网络获取中间层输出特征.

4) 将特征输入至检测网络, 输出检测结果.

Metzen 等人选取了 5 个中间层, 分别对应图 17 中检测输入 (0–4). 在 CIFAR-10 数据集上, 分别使用 FGSM、BIM 和 DeepFool 评估他们的算法, 分别达到了 91.6%、78.2% 和 75.4% 的平均检测准确率. 不同层作为输入的检测器对不同类型的攻击具有不同的检测效果. Metzen 等人[56]指出他们的方法之所以有效是因为攻击者需要同时找到使分类器和检测器出错的对抗扰动, 加大了攻击的难度. Carlini 等人[19]发现这种算法对于 CW 攻击算法具有更高的误检率, 并且可以通过替代攻击规避.

3.3.3 神经网络特性

3.3.3.1 SafetyNet

Lu 等人[50]介绍了 SafetyNet 检测方法, 它由常规 DNN 模型 (在他们的实验中为 VGG19[1]或 ResNet[70]) 和支持向量机 RBF-SVM[74]组成, 该 RBF-SVM 使用从最后一个 ReLU 层算出的离散编码来检测对抗样本, 它的检测框架如图 25 所示.

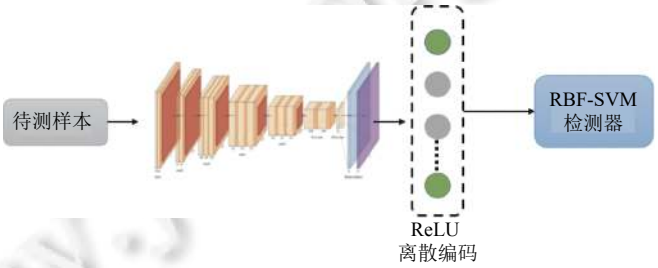


图 25 SafetyNet 检测框架

SafetyNet 将最后一个 ReLU 层在某一阈值集上量化以生成离散编码. 这种做法的前提是他们认为正常样本和对抗样本会出现不同的编码模式. SafetyNet 中的对抗样本检测器利用二进制码或四进制码 (激活模式) 的 RBF-SVM 来检测对抗样本.

用 c 代表编码, 则 RBF-SVM 通过以下方式分类:

$$SVM(c)=\sum_{i}^{N}\alpha_i y_i \exp\left(-\frac{\| c-c_i \|^2}{2\sigma^2}\right)+b \tag{42}$$

其中, σ 是 RBF-SVM 的参数. 当 σ 很小时, 检测器基本上不产生梯度, 除非对抗样本的编码 c 非常接近正常样本的编码 c_i.

SafetyNet 检测算法的具体检测步骤如下.

1) 将待测样本输入至原始 DNN 模型.

2) 对样本最后一个 ReLU 层进行量化计算得到离散编码.

3) 将该离散编码输入至 RBF-SVM 检测器, 得到检测结果.

Lu 等人使用 FGSM、BIM、DeepFool 攻击方法生成的对抗样本在 CIFAR-10 和 ImageNet 数据集上进行评估, 平均检测精度达到了 88.9%.

SafetyNet 虽然可以训练出难以欺骗的强大检测器, 但是它使用特定的攻击方法进行训练, 存在易受到未知攻击干扰的风险, 一个可能的原因是 SafetyNet 在训练时进行剪枝导致训练数据缺少未触及到的神经元、路径或激活模式, 因此对该方法对未知攻击的防范能力有待考证.

3.3.3.2 特征距离空间

Carrara 等人[51]受到文献 [75,76] 的启发, 对特征空间中特定图像的网络内部激活位置进行编码. 他们在网络

的每一个激活层定义了不同的特征距离空间, 统计了不同标签的正常样本在每个激活层网络的激活位置, 根据样本的标签划分区域, 并在这些区域中规定参考位置, 称为枢纽点. 给定输入样本 x, 计算 x 与网络内部激活空间的枢纽点之间的相对距离, 以此来区分正常样本与对抗样本 (通常对抗样本会偏离这些区域). 将输入 x 的所有激活层表示都嵌入到特征距离空间中, 就能通过网络的前向传递对激活的轨迹进行编码, 并且区分出正常样本与对抗样本所跟踪的激活轨迹之间的差异.

如图 26 所示为特征距离空间编码演化示意图, 图中每一层表示深度神经网络特定层激活模式定义的特征空间. 每个空间上圆表示特定类别的集群, 蓝色线条轨迹代表属于 3 个不同类别的正常样本, 红色轨迹代表对抗样本. 通过计算输入与特定类的一些参考点之间在特征空间距离来编码激活轨迹.

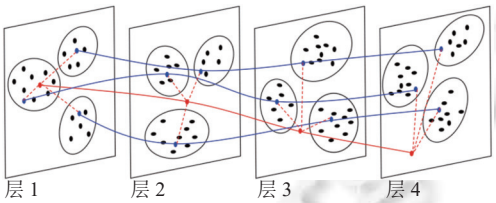


图 26 特征距离空间编码演化示意图

若深度神经网络 $F(x)$ 总共有 L 个激活层, 该数据集总共有 T 个类别, $o^{(l)}$ 为其第 l 层的激活输出, 对于每个激活层 l, 他们通过枢纽嵌入 (即在特征距离空间中进行嵌入) 对输出 $o^{(l)}$ 在特征空间中的位置进行编码, 其中每个维度代表与特定类别空间中的枢纽点之间的距离. 令 $\mathbf{P}^{(l)}=\{\mathbf{p}_1^{(l)},\ldots,\mathbf{p}_T^{(l)}\}$ 表示第 l 层的枢纽激活空间, $d(x,y)$ 为在实向量上定义的距离函数, 则输入 x 的枢纽嵌入 $e^{(l)}\in R^T$ 定义为:

$$\mathbf{e}^{(l)}=(d(\mathbf{o}^{(l)},\mathbf{p}_1^{(l)}),d(\mathbf{o}^{(l)},\mathbf{p}_2^{(l)}),\ldots,d(\mathbf{o}^{(l)},\mathbf{p}_T^{(l)})) \tag{43}$$

对于枢纽点 $\mathbf{P}^{(1)},\ldots,\mathbf{P}^{(L)}$ 的选择, 作者给出了两种方案. 方案 1 选择属于类别 t 的图像的第 l 层的激活质心作为枢纽:

$$\mathbf{p}_t^{(l)}=\frac{1}{|S_t|}\sum_{j=1}^{|S_t|}\mathbf{o}_{t,j}^{(l)} \tag{44}$$

其中, $|S_t|$ 表示类别为 t 的训练集样本的数量, 而 $\mathbf{o}_{t,j}^{(l)}$ 是由属于 t 类的第 j 个训练样本产生的第 l 层的激活.

方案 2 中, $\mathbf{p}_t^{(l)}$ 应满足自身与同一类别的所有其他样本之间的距离之和最小, 设 $O_t^{(l)}=\{\mathbf{o}_{t,1}^{(l)},\ldots,\mathbf{o}_{t,|S_t|}^{(l)}\}$ 属于 t 类的训练集样本的第 l 层的激活集, 则:

$$\mathbf{p}_t^{(l)}=\underset{x\in O_t^{(l)}}{\operatorname{argmin}}\sum_{j=1}^{|S_t|}\|x-\mathbf{o}_{t,j}^{(l)}\|_2 \tag{45}$$

特征距离空间检测方法的具体检测流程如下:

1) 给定输入样本 x.

2) 在神经网络前向传递的过程中计算输出 $(\mathbf{o}^{(1)},\ldots,\mathbf{o}^{(L)})$.

3) 计算每个维度的枢纽嵌入, 或者长度为 L 的 T 维的嵌入向量 $\mathbf{E}=(\mathbf{e}^{(1)},\ldots,\mathbf{e}^{(L)})$.

4) 将嵌入向量输入到预先训练好的 LSTM 二元分类检测器进行检测, 输出检测结果.

Carrara 等人在 ImageNet 大规模视觉识别挑战比赛 (ImageNet large scale visual recognition challenge, ILSVRC) 训练集上对该方法进行了评估, L-BFGS、FGSM、BIM 和 PGD 这 4 种攻击算法, 平均检测准确率分别为 61.7%、89.2%、93.3% 和 93.2%. 针对检测算法在 L-BFGS 上的表现, 作者认为是由于 L-BFGS 通常拥有更小的对抗扰动.

Carrara 等人[51]的工作给我们提供了一种新颖的思路, 他们灵活地利用了神经网络中的激活空间, 定义了特征距离的嵌入方式, 其方法中枢纽点的确定较为依赖训练数据, 枢纽点是否合理影响了检测算法的精度. 但是这种方法需要预先确定枢纽点, 获取内部输出, 无法独立于模型. 并且他们仅在单个数据集上进行了验证, 其泛化性还有待检验.

4 挑战与展望

纵观本文提及的对抗样本检测方面的工作, 该领域的研究仍然面临许多挑战和问题, 我们将在本节介绍这些问题.

1) 增强泛化性. 大多数检测算法的泛化性较差, 检测算法具有泛化性要求它能在各种不同攻击算法 (黑盒、白盒) 下保持稳定的检测水平, 而不是只对某一类对抗样本有检测能力. 这也与他们检测原理相关, 例如在有监督检测中, 检测器能够从被标记的正常样本和对抗样本学习, 通常拥有更好的性能. 但是有监督检测很大程度上只能检测已知攻击算法生成的对抗样本, 泛化性较差. 而无监督检测, 只通过正常样本构建检测器, 拥有更好的泛化性能. 但是它面临的两个主要挑战是: (1) 如何寻找正常样本不敏感而对抗样本敏感的区分特征; (2) 模型参数、阈值等区分特征非常难确定, 不能保证适用于所有模型和数据, 某些阈值可能仅在某种攻击算法下适用, 不同攻击算法需要进行试错调参. 想要提高检测器的泛化性, 一方面我们要考虑数据层面的区分特征, 另一方面要考虑神经网络的特性, 目前存在的检测算法大多割裂了两者的关系, 试图仅从一个方面实现样本的区别. 然后神经网络由于其训练拟合的特性, 它的最终模型参数其实和训练数据及其分布是息息相关的. 想要实现两者的结合, 一个可行的方案是使用联合检测方案, 不同检测方法各有优缺点, 使用联合检测的方法可以取长补短, 例如基于降噪的方法可以作为很多的方法的预处理阶段, 与特征距离空间等基于神经网络特征的技术结合, 每种方法根据其特点分配权重、或采用投票制的方式实现更加全面的检测.

2) 联合防御技术. 对抗样本防御技术旨在对输入的样本进行正确分类, 无论输入是正常样本还是对抗样本, 而检测算法可以区分两者. 因此, 可以结合两种技术, 实现更加全面完善的安全防护, 联合防护框架如图 27 所示. 当待测样本所添加的对抗扰动较小, 检测器无法感知时, 会通过检测器来到防御部分, 而防御模型通常具有较好的鲁棒性, 可以对细微扰动的对抗样本实现正确分类; 若攻击者想要实现对防御模型的攻击, 就需要增加对抗扰动, 而大的对抗扰动无法通过检测. 在检测算法与防御方法结合的机制下攻击者的攻击难度大大增加, 例如在第 3.2.2.2 节中所介绍的 MagNet[34]则是采用该种框架形式, 在待测样本通过检测器后还会通过“改良器”, “改良器”相当于框架中的防御模型, 实现对样本的正确分类. 目前检测算法和防御算法也已经有了一定的技术积累, 设计并结合更好的检测-防御联合防护框架将是未来一个较为有潜力的研究方向.

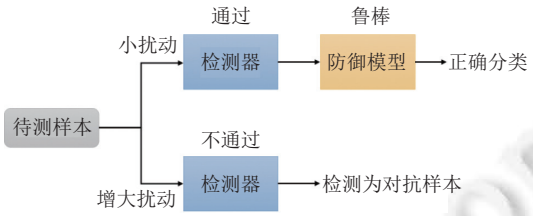


图 27 检测-防御联合防护框架

3) 轻量化. 构建检测器时增加考虑复杂度、运行资源、推理速度. 许多搭载轻量级神经网络的穿戴设备已经融入我们生活的各个角落, 例如具备人脸识别的智能手表. 这类设备通常不具备很强的算力, 且运行资源也有限, 要在这种设备上实现检测, 保护我们的隐私与安全, 就要求检测器在保持检测精度的同时占用更少的资源, 拥有较快推理速度. 而本文中所介绍的大部分检测算法, 均在实验室环境下利用算力强大的 GPU 服务器集群实现, 并没有针对现实应用场景进行相应的资源优化, 该问题会阻碍对抗样本检测走向实际应用. 因此, 我们在投入研究的时候也应考虑到实际应用场景的需求, 设计更加轻量化的检测算法.

4) 研究对抗样本. 虽然在绝大多数场景下, 对抗样本扮演“坏人”的角色, 但是 Ilyas 等人[56]的最新研究认为: “对抗样本不是 bug, 而是特征”. 他们将对抗样本的现象归结为标准 ML 数据集中存在高度预测性但非鲁棒性的自然结果, 他们在实验中区分数据集中的鲁棒特征和非鲁棒特征, 并证明仅非鲁棒特征就能实现良好的特征概括. 因此, 研究对抗样本能帮助我们对输入数据的特征有更深的了解, 足够多的数据能够帮助我们构建更鲁棒的深度学习模型和泛化性更好的检测器, 保护我们的数据隐私.

5) 公开源码及数据. 很多对抗样本检测方法没有公开其源代码或者实验设置不够清晰, 研究人员复现算法会耗费大量的精力. 虽然公开源代码并不是一项义务, 但是我们还是建议研究者将源代码公开, 这样既可以促进该领域的研究, 也能通过开源社区的力量帮助研究者改进算法.

5 结 论

深度神经网络的安全问题是目前人工智能技术走向应用的关键一环, 人工智能对抗攻防技术的发展显得尤为关键, 攻防不断博弈进步, 而对抗样本检测作为防御的一种手段, 近几年来得到了越来越多研究者的关注且发展迅速. 从 Hendrycks 和 Gimpel[30]的早期理论探索, 到神经元激活状态、特征对齐等, 不断有新的思路注入这个领域. 对抗样本检测还借鉴了其他领域的很多方法来完善自身, 例如一些方法的数据处理用到了许多经典数字图像处理的方法, 局部内在维度的引入 (起初被应用于异常检测), 也有的方法将对抗式生成网络引入到检测任务中, 对抗样本检测是一个融合多元知识的交叉领域. 本文首次综合性地介绍了目前图像分类领域的深度神经网络对抗样本检测方法, 从原理层面对各种方法进行了分类, 并简要地分析了检测方法的理论和特点. 最后希望通过本文为未来图像分类领域对抗样本检测的研究提供帮助.

References:

[1] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: Proc. of the 3rd Int'l Conf. on Learning Representations. San Diego, 2015. 1–14.

[2] Szegedy C, Liu W, Jia YQ, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. In: Proc. of the 2015 IEEE Conf. on Computer Vision and Pattern Recognition. Boston: IEEE, 2015. 1–9. [doi: 10.1109/CVPR.2015.7298594]

[3] Huang PS, Kim M, Hasegawa-Johnson M, Smaragdis P. Joint optimization of masks and deep recurrent neural networks for monaural source separation. IEEE/ACM Trans. on Audio, Speech, and Language Processing, 2015, 23(12): 2136–2147. [doi: 10.1109/TASLP.2015.2468583]

[4] Williamson DS, Wang YX, Wang DL. Complex ratio masking for monaural speech separation. IEEE/ACM Trans. on Audio, Speech, and Language Processing, 2016, 24(3): 483–492. [doi: 10.1109/TASLP.2015.2512042]

[5] Zeng Y, Zhang M, Han F, Gong Y, Zhang J. Spectrum analysis and convolutional neural network for automatic modulation recognition. IEEE Wireless Communications Letters, 2019, 8(3): 929–932. [doi: 10.1109/LWC.2019.2900247]

[6] Xu JL, Luo CB, Parr G, Luo Y. A spatiotemporal multi-channel learning framework for automatic modulation recognition. IEEE Wireless Communications Letters, 2020, 9(10): 1629–1632. [doi: 10.1109/LWC.2020.2999453]

[7] Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. In: Proc. of the 5th Int'l Conf. on Learning Representations. Toulon: OpenReview.net, 2017.

[8] Abu-El-Haija S, Perozzi B, Kapoor A, Alipourfard N, Lerman K, Harutyunyan H, Steeg GV, Galstyan A. MixHop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In: Proc. of the 36th Int'l Conf. on Machine Learning. Long Beach: PMLR, 2019. 21–29.

[9] Eykholt K, Evtimov I, Fernandes E, Li B, Rahmati A, Xiao CW, Prakash A, Kohno T, Song D. Robust physical-world attacks on deep learning visual classification. In: Proc. of the 2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 1625–1634. [doi: 10.1109/CVPR.2018.00175]

[10] Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. In: Proc. of the 3rd Int'l Conf. on Learning Representations. San Diego, 2015. 1–11.

[11] Gu SX, Rigazio L. Towards deep neural network architectures robust to adversarial examples. In: Proc. of the 3rd Int'l Conf. on Learning Representations. San Diego, 2015. 1–9.

[12] Papernot N, McDaniel P, Wu X, Jha S, Swami A. Distillation as a defense to adversarial perturbations against deep neural networks. In: Proc. of the 2016 IEEE Symp. on Security and Privacy (SP). San Jose: IEEE, 2016. 582–597. [doi: 10.1109/SP.2016.41]

[13] Carlini N, Wagner D. Defensive distillation is not robust to adversarial examples. arXiv:1607.04311, 2016.

[14] Akhtar N, Mian A. Threat of adversarial attacks on deep learning in computer vision: A survey. IEEE Access, 2018, 6: 14410–14430. [doi: 10.1109/ACCESS.2018.2807385]

[15] Wang XM, Li J, Kuang XH, Tan YA, Li J. The security of machine learning in an adversarial setting: A survey. Journal of Parallel and

Distributed Computing, 2019, 130: 12–23. [doi: 10.1016/j.jpdc.2019.03.003]

[16] Xu H, Ma Y, Liu HC, Deb D, Liu H, Tang JL, Jain AK. Adversarial attacks and defenses in images, graphs and text: A review. Int'l Journal of Automation and Computing, 2020, 17(2): 151–178. [doi: 10.1007/s11633-019-1211-x]

[17] Bulusu S, Kailkhura B, Li B, Varshney P, Song D. Anomalous instance detection in deep learning: A survey. Livermore: Lawrence Livermore National Laboratory, 2020. 1–20.

[18] Miller D, Wang YJ, Kesidis G. When not to classify: Anomaly detection of attacks (ADA) on DNN classifiers at test time. Neural Computation, 2019, 31(8): 1624–1670. [doi: 10.1162/neco_a_01209]

[19] Carlini N, Wagner D. Adversarial examples are not easily detected: Bypassing ten detection methods. In: Proc. of the 10th ACM Workshop on Artificial Intelligence and Security. Dallas: ACM, 2017. 3–14. [doi: 10.1145/3128572.3140444]

[20] Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow IJ, Fergus R. Intriguing properties of neural networks. In: Proc. of the 2nd Int'l Conf. on Learning Representations. Banff, 2014. 1–10.

[21] Kurakin A, Goodfellow IJ, Bengio S. Adversarial machine learning at scale. In: Proc. of the 5th Int'l Conf. on Learning Representations. Toulon: OpenReview.net, 2017.

[22] Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A. Towards deep learning models resistant to adversarial attacks. In: Proc. of the 6th Int'l Conf. on Learning Representations. Vancouver: OpenReview.net, 2018.

[23] Carlini N, Wagner D. Towards evaluating the robustness of neural networks. In: Proc. of the 2017 IEEE Symp. on Security and Privacy (SP). San Jose: IEEE, 2017. 39–57. [doi: 10.1109/SP.2017.49]

[24] Papernot N, McDaniel P, Jha S, Fredrikson M, Celik ZB, Swami A. The limitations of deep learning in adversarial settings. In: Proc. of the 2016 IEEE European Symp. on Security and Privacy (Euro S&P). Saarbruecken: IEEE, 2016. 372–387. [doi: 10.1109/EuroSP.2016.36]

[25] Moosavi-Dezfooli SM, Fawzi A, Frossard P. DeepFool: A simple and accurate method to fool deep neural networks. In: Proc. of the 2016 IEEE Conf. on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 2574–2582. [doi: 10.1109/CVPR.2016.282]

[26] Deng L. The MNIST database of handwritten digit images for machine learning research [Best of the Web]. IEEE Signal Processing Magazine, 2012, 29(6): 141–142. [doi: 10.1109/MSP.2012.2211477]

[27] Krizhevsky A, Hinton GE. Learning multiple layers of features from tiny images. 2009. https://www.cs.utoronto.ca/~kriz/learning-features-2009-TR.pdf

[28] Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma SA, Huang ZH, Karpathy A, Khosla A, Bernstein M, Berg AC, Fei-Fei L. ImageNet large scale visual recognition challenge. Int'l Journal of Computer Vision, 2015, 115(3): 211–252. [doi: 10.1007/s11263-015-0816-y]

[29] Netzer Y, Wang T, Coates A, Bissacco A, Wu B, Ng AY. Reading digits in natural images with unsupervised feature learning. In: Proc. of the 2011 NIPS Workshop on Deep Learning and Unsupervised Feature Learning. MIT Press, 2011. 1–9.

[30] Hendrycks D, Gimpel K. Early methods for detecting adversarial images. In: Proc. of the 5th Int'l Conf. on Learning Representations. Toulon: OpenReview.net, 2017.

[31] Carrara F, Falchi F, Caldelli R, Amato G, Becarelli R. Adversarial image detection in deep neural networks. Multimedia Tools and Applications, 2019, 78(3): 2815–2835. [doi: 10.1007/s11042-018-5853-4]

[32] Wang JY, Dong GL, Sun J, Wang XY, Zhang PX. Adversarial sample detection for deep neural network through model mutation testing. In: Proc. of the 41st Int'l Conf. on Software Engineering (ICSE). Montreal: IEEE, 2019. 1245–1256. [doi: 10.1109/ICSE.2019.00126]

[33] Xu WL, Evans D, Qi YJ. Feature squeezing: Detecting adversarial examples in deep neural networks. In: Proc. of the 25th Annual Network and Distributed System Security Symp. San Diego: The Internet Society, 2018. 1–16.

[34] Meng DY, Chen H. MagNet: A two-pronged defense against adversarial examples. In: Proc. of the 2017 ACM SIGSAC Conf. on Computer and Communications Security. Dallas: ACM, 2017. 135–147. [doi: 10.1145/3133956.3134057]

[35] Liang B, Li HC, Su MQ, Li XR, Shi WC, Wang XF. Detecting adversarial image examples in deep neural networks with adaptive noise reduction. IEEE Trans. on Dependable and Secure Computing, 2021, 18(1): 72–85. [doi: 10.1109/TDSC.2018.2874243]

[36] Creswell A, White T, Dumoulin V, Arulkumaran K, Sengupta B, Bharath AA. Generative adversarial networks: An overview. IEEE Signal Processing Magazine, 2018, 35(1): 53–65. [doi: 10.1109/MSP.2017.2765202]

[37] Samangouei P, Kabkab M, Chellappa R. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In: Proc. of the 6th Int'l Conf. on Learning Representations. Vancouver: OpenReview.net, 2018.

[38] Akcay S, Atapour-Abarghouei A, Breckon TP. GANomaly: Semi-supervised anomaly detection via adversarial training. In: Proc. of the 14th Asian Conf. on Computer Vision. Perth: Springer, 2019. 622–637. [doi: 10.1007/978-3-030-20893-6_39]

[39] Zheng ZH, Hong PY. Robust detection of adversarial attacks by modeling the intrinsic properties of deep neural networks. In: Proc. of the

32nd Int'l Conf. on Neural Information Processing Systems. Montreal: Curran Associates Inc., 2018. 7924–7933.

[40] Shumailov I, Zhao YR, Mullins R, Anderson R. The taboo trap: Behavioural detection of adversarial samples. arXiv:1811.07375, 2018.

[41] Freitas S, Chen ST, Wang ZJ, Chau DH. UnMask: Adversarial detection and defense through robust feature alignment. In: Proc. of the 2020 IEEE Int'l Conf. on Big Data (Big Data). Atlanta: IEEE, 2020. 1081–1088. [doi: 10.1109/BigData50022.2020.9378303]

[42] Feinman R, Curtin RR, Shintre S, Gardner AB. Detecting adversarial samples from artifacts. arXiv:1703.00410, 2017.

[43] Ma XJ, Li B, Wang YS, Erfani SM, Wijewickrema SNR, Schoenebeck G, Song D, Houle ME, Bailey J. Characterizing adversarial subspaces using local intrinsic dimensionality. In: Proc. of the 6th Int'l Conf. on Learning Representations. Vancouver: OpenReview.net, 2018. 1–15.

[44] Cohen G, Sapiro G, Giryes R. Detecting adversarial samples using influence functions and nearest neighbors. In: Proc. of the 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020. 14441–14450. [doi: 10.1109/CVPR42600.2020.01446]

[45] Lee S, Kim NR, Cho Y, Choi JY, Kim S, Kim JA, Lee JH. Adversarial detection with Gaussian process regression-based detector. KSII Trans. on Internet and Information Systems, 2019, 13(8): 4285–4299. [doi: 10.3837/tiis.2019.08.027]

[46] Grosse K, Manoharan P, Papernot N, Backes M, McDaniel P. On the (statistical) detection of adversarial examples. arXiv:1702.06280, 2017.

[47] Gong ZT, Wang WL, Ku WS. Adversarial and clean data are not twins. arXiv:1704.04960, 2017.

[48] Yin XW, Kolouri S, Rohde GK. GAT: Generative adversarial training for adversarial example detection and robust classification. In: Proc. of the 8th Int'l Conf. on Learning Representations. Addis Ababa: OpenReview.net, 2020.

[49] Metzen JH, Genewein T, Fischer V, Bischoff B. On detecting adversarial perturbations. In: Proc. of the 5th Int'l Conf. on Learning Representations. Toulon: OpenReview.net, 2017.

[50] Lu JJ, Issaranon T, Forsyth D. SafetyNet: Detecting and rejecting adversarial examples robustly. In: Proc. of the 2017 IEEE Int'l Conf. on Computer Vision. Venice: IEEE, 2017. 446–454. [doi: 10.1109/ICCV.2017.56]

[51] Carrara F, Becarelli R, Caldelli R, Falchi F, Amato G. Adversarial examples detection in features distance spaces. In: Proc. of the 2019 European Conf. on Computer Vision (ECCV) Workshops. Munich: Springer, 2019. 313–327. [doi: 10.1007/978-3-030-11012-3_26]

[52] Kullback S, Leibler RA. On information and sufficiency. The Annals of Mathematical Statistics, 1951, 22(1): 79–86. [doi: 10.1214/aoms/1177729694]

[53] Wiyatno RR, Xu AQ, Dia O, de Berker A. Adversarial examples in modern machine learning: A review. arXiv:1911.05268, 2019.

[54] Buades A, Coll B, Morel JM. Non-local means denoising. Image Processing on Line, 2011, 1: 208–212. [doi: 10.5201/ipol.2011.bcm_nlm]

[55] Zagoruyko S, Komodakis N. Wide residual networks. In: Proc. of the 2016 British Machine Vision Conf. York: BMVA Press, 2016. 1–14.

[56] Ilyas A, Santurkar S, Tsipras D, Engstrom L, Tran B, Mądry A. Adversarial examples are not bugs, they are features. In: Proc. of the 33rd Int'l Conf. on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2019. 12.

[57] Tsipras D, Santurkar S, Engstrom L, Turner A, Madry A. There is no free lunch in adversarial robustness (but there are unexpected benefits). arXiv:1805.12152, 2018.

[58] He KM, Gkioxari G, Dollár P, Girshick R. Mask R-CNN. In: Proc. of the 2017 IEEE Int'l Conf. on Computer Vision. Venice: IEEE, 2017. 2980–2988. [doi: 10.1109/ICCV.2017.322]

[59] Zhang YT, Lee K, Lee H. Augmenting supervised neural networks with unsupervised objectives for large-scale image classification. In: Proc. of the 33rd Int'l Conf. on Machine Learning. New York: JMLR.org, 2016. 612–621.

[60] Bengio Y, Mesnil G, Dauphin Y, Rifai S. Better mixing via deep representations. In: Proc. of the 30th Int'l Conf. on Machine Learning. Atlanta: JMLR.org, 2013. I-552–I-560.

[61] Gardner JR, Upchurch P, Kusner MJ, Li YX, Weinberger KQ, Bala K, Hopcroft JE. Deep manifold traversal: Changing labels with convolutional features. arXiv:1511.06421, 2015.

[62] Gal Y, Ghahramani Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: Proc. of the 33rd Int'l Conf. on Machine Learning. New York: JMLR.org, 2016. 1050–1059.

[63] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 2014, 15(1): 1929–1958.

[64] LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD. Backpropagation applied to handwritten zip code recognition. Neural Computation, 1989, 1(4): 541–551. [doi: 10.1162/neco.1989.1.4.541]

[65] Houle ME. Local intrinsic dimensionality I: An extreme-value-theoretic foundation for similarity applications. In: Proc. of the 10th Int'l

Conf. on Similarity Search and Applications. Munich: Springer, 2017. 64–79. [doi: 10.1007/978-3-319-68474-1_5]

[66] Houle ME. Local intrinsic dimensionality II: Multivariate analysis and distributional support. In: Proc. of the 10th Int'l Conf. on Similarity Search and Applications. Munich: Springer, 2017. 80–95. [doi: 10.1007/978-3-319-68474-1_6]

[67] Amsaleg L, Chelly O, Furon T, Girard S, Houle ME, Kawarabayashi KI, Nett M. Estimating local intrinsic dimensionality. In: Proc. of the 21st ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. Sydney: ACM, 2015. 29–38. [doi: 10.1145/2783258.2783405]

[68] Lu PH, Chen PY, Yu CM. On the limitation of local intrinsic dimensionality for characterizing the subspaces of adversarial examples. In: Proc. of the 6th Int'l Conf. on Learning Representations. Vancouver: OpenReview.net, 2018.

[69] Koh PW, Liang P. Understanding black-box predictions via influence functions. In: Proc. of the 34th Int'l Conf. on Machine Learning. Sydney: JMLR.org, 2017. 1885–1894.

[70] He KM, Zhang XY, Ren SQ, Sun J. Deep residual learning for image recognition. In: Proc. of the 2016 IEEE Conf. on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 770–778. [doi: 10.1109/CVPR.2016.90]

[71] Ebden M. Gaussian processes: A quick introduction. arXiv:1505.02965, 2015.

[72] Nickisch H, Rasmussen CE. Approximations for binary Gaussian process classification. Journal of Machine Learning Research, 2008, 9(67): 2035–2078.

[73] Snoek J, Larochelle H, Adams RP. Practical Bayesian optimization of machine learning algorithms. In: Proc. of the 25th Int'l Conf. on Neural Information Processing Systems. Lake Tahoe: Curran Associates Inc., 2012. 2951–2959.

[74] Eigen D, Puhrsch C, Fergus R. Depth map prediction from a single image using a multi-scale deep network. In: Proc. of the 27th Int'l Conf. on Neural Information Processing Systems. Montreal: MIT Press, 2014. 2366–2374.

[75] Zezula P, Amato G, Dohnal V, Batko M. Similarity Search: The Metric Space Approach. New York: Springer, 2006.

[76] Connor R, Vadicamo L, Rabitti F. High-dimensional simplexes for supermetric search. In: Proc. of the 10th Int'l Conf. on Similarity Search and Applications. Munich: Springer, 2017. 96–109.

周涛(1997－), 男, 硕士, 主要研究领域为对抗攻击, 对抗样本检测.

王竟亦(1991－), 男, 博士, 研究员, 博士生导师, 主要研究领域为智能系统安全, 形式化方法, 软件工程.

甘燃(1997－), 男, 硕士, 主要研究领域为对抗攻击, 对抗样本检测.

宣琦(1981－), 男, 博士, 教授, 博士生导师, 主要研究领域为人工智能安全, 智能信号分析, 网络数据挖掘.

徐东伟(1985－), 男, 博士, 副教授, 主要研究领域为人工智能应用及安全.