

面向深度学习模型的可靠性测试综述

陈若曦¹, 金海波¹, 陈晋音^{1,2}, 郑海斌^{1,2}, 李晓豪¹

¹浙江工业大学信息工程学院 杭州 310023

²浙江工业大学网络空间安全研究院 杭州 310023

摘要 深度学习模型由于其出色的性能表现而在各个领域被广泛应用,但它们在面对不确定输入时,往往会出现意料之外的错误行为,在诸如自动驾驶系统等安全关键应用,可能会造成灾难性的后果。深度模型的可靠性问题引起了学术界和工业界的广泛关注。因此,在深度模型部署前迫切需要对模型进行系统性测试,通过生成测试样本,并由模型的输出得到测试报告,以评估模型的可靠性,提前发现潜在缺陷。一大批学者分别从不同测试目标出发,对模型进行测试,并且提出了一系列测试方法。目前对测试方法的综述工作只关注到模型的安全性,而忽略了其他测试目标,且缺少对最新出版的方法的介绍。因此,本文拟对模型任务性能、安全性、公平性和隐私性4个方面对现有测试技术展开全方位综述,对其进行全面梳理、分析和总结。具体而言,首先介绍了深度模型测试的相关概念;其次根据不同测试目标对79篇论文中的测试方法和指标进行分类介绍;然后总结了目前深度模型可靠性测试在自动驾驶、语音识别和自然语言处理三个工业场景的应用,并提供了可用于深度模型测试的24个数据集、7个在线模型库和常用工具包;最后结合面临的挑战和机遇,对深度模型可靠性测试的未来研究方向进行总结和展望,为构建系统、高效、可信的深度模型测试研究提供参考。值得一提的是,本文将涉及的数据集、模型、测试方法代码、评价指标等资料归纳整理在<https://github.com/Allen-piexl/Testing-Zoo>,方便研究人员下载使用。

关键词 深度学习模型;深度测试;可靠性;安全性;公平性;隐私性

中图法分类号 TP391 DOI号 10.19363/J.cnki.cn10-1380/tn.2024.01.03

Deep Learning Testing for Reliability: A Survey

CHEN Ruoxi¹, JIN Haibo¹, CHEN Jinyin^{1,2}, ZHENG Haibin^{1,2}, LI Xiaohao¹

¹ College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China

² College of Institute of Cyberspace Security, Zhejiang University of Technology, Hangzhou 310023, China

Abstract Deep neural networks (DNNs) have been widely applied in various areas due to impressive capabilities and outstanding performance. However, they will expose unexpected erroneous behaviors when they are faced with uncertainty, which may lead to disastrous consequences in safety-critical applications such as autonomous driving systems. The reliability of deep models has aroused widespread concern in both academia and industry. Therefore, it is necessary to systematically test deep models before the deployment. The reliability of models can be evaluated and potential defects can be found in advance by generating testing examples and then obtaining test reports from the output of models. A large number of researchers have conducted in-depth research on testing DNNs and proposed a series of testing methods from different testing objectives. However, current works on survey of testing methods only focus on the security of DNNs, and they don't take recently-published techniques into consideration. Different from them, this article focuses on four reliability test objectives of models, i.e., task performance, security, fairness and privacy, and comprehensively analyzes the related technologies and methods of testing DNNs. Firstly, the related concepts of deep learning testing are introduced. Then, according to different testing objectives, testing methods and metrics from 79 papers are classified and introduced in detail. Next, the current application of DNNs' reliability testing in three industrial scenarios are summarized, including autonomous driving, speech recognition and natural language processing. Besides, 24 datasets, 7 online model libraries and common toolkits that can be used for deep model testing are provided. Finally, along with the challenges and opportunities, the future research direction of deep learning testing is summarized, which provides reference for the construction of systematic, efficient and reliable deep learning testing. It is worth noting that the related datasets, open-source code of testing methods and metrics are available in <https://github.com/Allen-piexl/Testing-Zoo>, to facilitate subsequent scholars' research.

Key words deep neural networks; deep testing; reliability; security; fairness; privacy

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

本课题得到国家自然科学基金(No. 62072406)、信息系统安全技术重点实验室基金(No. 61421110502)、国家重点研发计划基金资助项目(No. 2018AAA0100801)、国家自然科学基金项目-联合重点(No. U21B2001)、浙江省重点研发计划项目(No. 2022C01018)资助。

收稿日期: 2022-04-06; 修改日期: 2022-06-22; 定稿日期: 2023-09-26

1 引言

人工智能作为引领数字化未来的战略性技术, 日益成为驱动经济社会各领域加速跃升的重要引擎。近年来, 数据量爆发式增长、计算能力显著性提升、深度学习算法突破性应用, 极大地推动了人工智能发展。然而, 深度神经网络(Deep Neural Networks, DNNs)潜在的安全问题也给智能驱动的数字世界带来了极大的安全隐患。DNNs 在面对不确定输入时, 往往会出现意料之外的错误行为, 在诸如自动驾驶系统^[1]、机器翻译^[2]和医疗^[3]等安全关键应用, 可能会造成灾难性的后果, 阻碍深度模型的实际部署。例如, 谷歌的自动驾驶汽车撞上了一辆公共汽车, 只是由于公共汽车没有按照它预计的情况让行^①。一辆自动驾驶的特斯拉汽车撞上了一辆拖车, 因为其自动驾驶系统无法将拖车识别为障碍物^[4]。深度学习模型在应用时的可靠性问题引起了学术界和工业界的广泛关注。

在正常工作时, 深度学习模型能输出正确且无偏见的结果, 面对恶意输入具有一定的抵抗能力, 同时对隐私信息具有保护能力, 是模型可靠应用的基本要求。为了提前发现模型中的潜在缺陷以避免事故发生, 亟需在部署前对其进行系统性测试。通过生成大量多样性测试样本, 并由模型的输出得到测试报告, 面向深度模型的测试技术能够在早期阶段对模型进行可靠性评估和潜在缺陷的检测, 降低模型在运行过程中发生错误的概率, 并有助于提高深度学习模型的可靠性。在现实中, 如何全面且高效地对深度模型进行测试, 实现极端情况的评估和监督, 进一步提高模型应用的可靠性, 成为安全可靠人工智能研究中的一个关键问题。

可靠的模型需要满足分类准确、风险鲁棒、决策无偏、隐私安全的基本条件。研究者们从不同角度出发, 提出了多种面向深度模型可靠性的测试方法。根据不同的模型属性, 可靠性测试的目标具体可分为向模型任务性能、安全性、公平性和隐私性四大类。模型的任务性能主要包括分类准确率和训练程度, 安全性指的是模型抵御外在风险的能力。模型的公平性衡量了模型输出结果的公平程度, 隐私性则反映了模型对用户数据的保护能力。模型的任务性能越好, 安全性和公平性越好, 对隐私的保护能力越强, 它的可靠性就越高, 在应用时出现错误分类的概率也就越低。这些测试属性是深度模型行为

的不同外部表现, 均在一定程度上决定了模型的行为和使用时的可靠性。

针对深度学习模型的测试方法, 已有工作^[5-7]对测试相关技术进行了分析和总结。但这些工作仅关注模型的安全性测试方法, 而忽略了对其他目标的测试, 且缺乏对 2020 年后新出版的研究的介绍。本文从深度学习模型任务性能、安全性、公平性和隐私性四个方面对已有的可靠性测试方法进行全面梳理、分析和总结, 为其可靠应用提供系统性综述。将目前研究成果按测试目标来分类, 总结成如图 1 所示的测试目标研究分布图。在所调研的 79 篇与测试方法相关的国内外文献中, 72%(57 篇)都关注了模型的安全性, 而关注其他可靠性测试目标的研究仅占少数。这证明了深度学习模型的安全性研究已成为学术界的热门方向, 也反映出目前深度模型测试研究目标的不平衡。事实上, 模型的公平性和隐私性也是值得测试研究的重要方向。

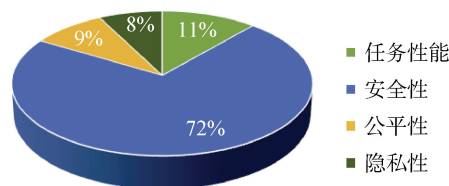


图 1 测试目标研究分布

Figure 1 The distribution of testing objectives

根据潜在威胁的作用阶段, 模型的安全性可以细分为推理阶段的对抗性输入^[8-11]和训练阶段容易受到污染数据而留下中毒后门^[12-15]。据此, 面向模型安全性的测试方法可以分为两类: 面向推理阶段和训练阶段安全性的测试方法。

在推理阶段, 测试方法可以分为基于覆盖率的测试方法和基于变异的测试方法, 其中覆盖率指标借鉴了软件测试中代码覆盖率的概念, 用于衡量测试的充分性。但由于深度模型和代码的显著差异, 它也存在一定的局限。基于变异的测试方法通过计算变异分数来发现模型中隐藏的缺陷。此外, 在得到模型的测试报告后, 基于修复的方法生成样本对模型进行测试和修复, 以进一步提高其鲁棒性和分类精度。

面向训练阶段安全性的测试方法可以分为离线和在线检测, 它们通过不同的方式观察模型的异常行为, 确定模型是否存在中毒后门。

此外, 由于深度模型输入维数高, 内部潜在特征空间大, 需要大量的测试样本对其进行测试, 也

① 相关报导见下: <http://www.theverge.com/2016/2/29/11134344/google-selfdriving-car-crash-report>

需要大量人力成本对样本进行标记。面向测试样本的选取方法对大量待标记的测试样本进行优先级排序, 以降低标记成本, 提升测试效率。

综上, 为了更好地探究该领域的测试方法以及相关进展, 本文对目前的研究成果进行梳理和总结, 具体的思维脉络如图 2 所示。

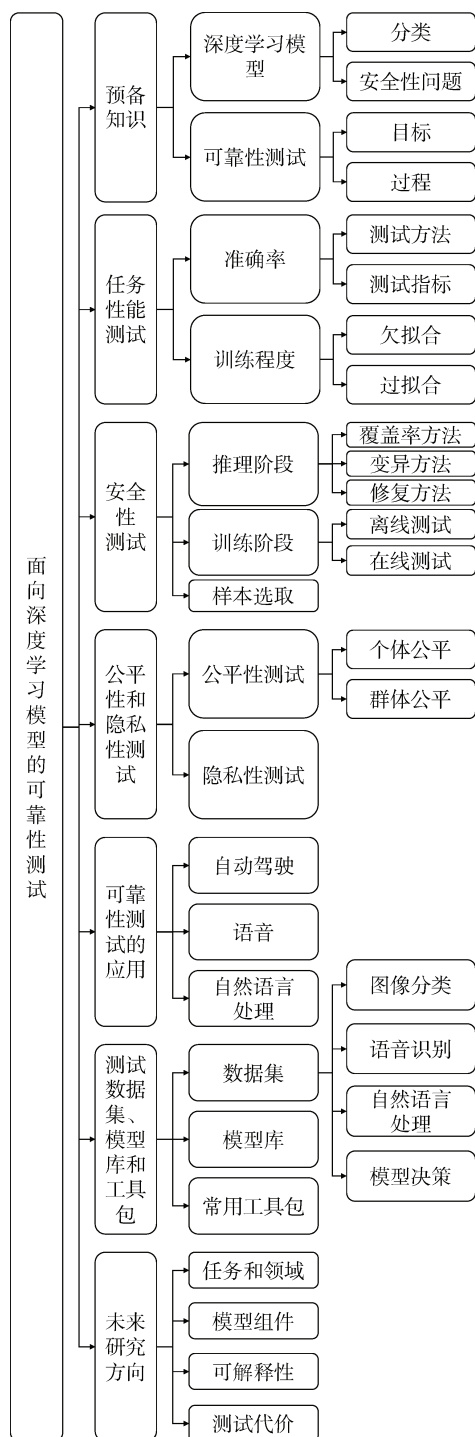


图 2 本文主要思维脉络

Figure 2 The framework of this paper

具体而言, 本文在第 2 节介绍了深度模型和安全性问题, 具体介绍了可靠性测试的相关概念; 第 3~5 节分别从模型任务性能、安全性、公平性和隐私性四个不同的角度出发, 对目前的测试方法进行了详细地介绍与分析; 第 6 节具体介绍了可靠性测试在多个领域中的应用; 第 7 节阐述目前不同领域中适用于测试的数据集, 此外还介绍了不同的模型库和常用工具包; 最后总结了面向深度模型可靠性测试的未来发展方向。本文涉及的开源代码和论文资源已经整理到公开的 GitHub 仓库^①, 方便后续学者进行研究。

2 可靠性测试基本概念

本节将分别对深度学习模型及其可靠性测试的基本概念进行定义。

2.1 深度学习模型

深度神经网络(Deep Neural Network, DNN)是最具代表性的深度学习模型之一, 仅需要用标记的训练数据进行训练, 就能够从原始输入中自动识别和提取相关的高级特征。

DNN 由多层组成, 每层包含多个神经元, 如图 3 所示。神经元是 DNN 中的一个独立计算单元, 它对其输入应用激活函数, 并将结果传递给其他连接的神经元。DNN 通常包括一个输入层、一个输出层和一个或多个隐藏层。层中的每个神经元都与下一层中的神经元有直接连接。总体而言, DNN 可以在数学上定义为多输入、多输出的参数函数, 由代表不同神经元的多个参数子函数组成。在运行过程中, 每一层都将其输入中包含的信息转换为更高级别的数据表示。

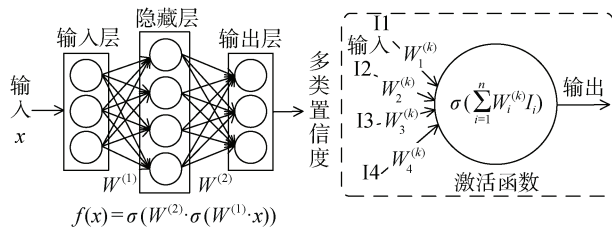


图 3 深度学习模型示意图^[16]

Figure 3 The architecture of DNN^[16]

2.1.1 深度学习模型分类

深度学习模型主要分为卷积神经网络(Convolutional Neural Network, CNN)和循环神经网络(Recurrent Neural Network, RNN)。

① <https://github.com/Allen-piexl/Testing-Zoo>

卷积神经网络。卷积神经网络是包含卷积计算且具有深度结构的前馈神经网络, 它适合处理具有类似网格结构的数据, 目前已广泛应用于图像分类和自然语言处理。CNN 主要由卷积层、池化层和全连接层组成, 分别用于提取特征、选择特征和分类回归。通过局部连接和全局共享, 能够学习大量的输入与输出之间的映射关系, 简化了模型复杂度, 减少了模型的参数。

循环神经网络。循环神经网络是一类用于处理序列数据的神经网络, 能以很高的效率对序列的非线性特征进行学习。简单的 RNN 结构如图 4 所示。它是一个由类似神经元的节点组成的网络, 它将数据流和维护的内部状态向量作为输入。RNN 在数据到达时处理一小块数据, 并在每次迭代中依次产生输出, 同时更新内部状态。RNN 中的信息不仅从前一层神经层流向后一层神经层, 而且还会从当前层状态迭代中流向后续层神经层。RNN 的状态特性有助于其在处理序列数据(例如音频和文本)方面取得巨大成功。目前, 长短期记忆网络(Long Short-Term Memory, LSTM)和门控循环单元(Gate Recurrent Unit, GRU)是最先进且使用最广泛的 RNN。

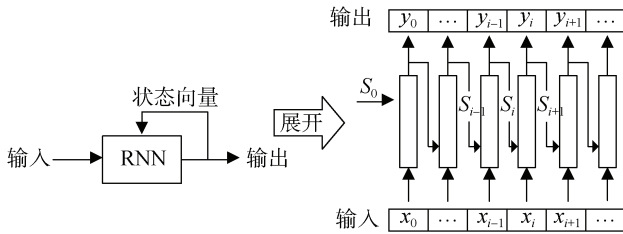


图 4 RNN 模型示意图

Figure 4 The architecture of RNN

2.1.2 深度模型的安全性问题

深度学习模型在应用时面临多种风险, 其中最为致命的则是不确定输入带来的安全性问题, 这也是可靠性测试学术界中最为重视的测试目标。本节将对其进行重点介绍。

深度学习模型往往容易受到不确定输入的影响而出现误判断。根据安全性威胁作用的不同阶段, 它们可以分为推理阶段的对抗性攻击和训练阶段的中毒攻击。

对抗攻击。对抗攻击发生在模型的推理阶段, 通过在正常样本中添加微小的扰动造成模型的错误输出, 如图 5 所示。

给定训练完全的模型 $f: \mathcal{X} \rightarrow \mathcal{Y}$ 和正常样本 $x \in \mathcal{X}$, 应有 $f(x; \theta) = y^c, y^c \in \mathcal{Y}$, 其中 y^c 表示模型

的预测输出, c 表示样本 x 所属真实类标, θ 表示模型的训练参数。该模型的一个对抗样本 x^* , 通常是由良性输入 x 增加扰动生成的, 它使模型分类错误。攻击的形式分为无目标攻击和有目标攻击, 对于无目标攻击, 优化目标为:

$$x^* = \arg \max \ell(x; \theta), s.t. \|x - x^*\|_p \leq \varepsilon \quad (1)$$

对于有目标攻击, 优化目标则为:

$$x^* = \arg \min \ell(x, t; \theta), s.t. \|x - x^*\|_p \leq \varepsilon \quad (2)$$

其中, $\ell(\cdot)$ 表示损失函数, 它衡量了模型 f 的预测输出 $f(x; \theta)$ 与错误类标 t 之间的差异。 $\|\cdot\|_p$ 表示 l_p 范数距离, 且 $p \in \mathbb{N}$ 。 $\|x - x^*\|_p \leq \varepsilon$ 保证了对抗样本和良性样本在语义上的相似性, 使攻击更为隐蔽。

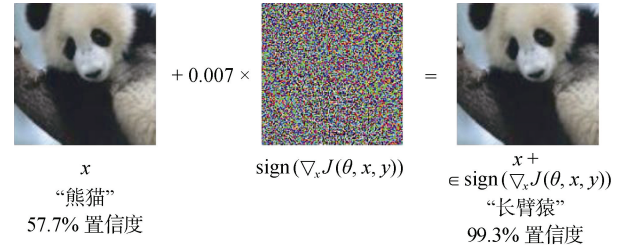


图 5 对抗攻击后, 模型以很高的置信度将熊猫样本识别成长臂猿^[8]

Figure 5 The image of panda is misclassified as gibbons with high confidence after adversarial attacks^[8]

中毒攻击。中毒攻击发生在深度学习模型的训练阶段, 通过在训练样本中注入中毒样本, 使得训练好的模型中存在后门。该模型平时正常工作, 而当携带触发标志的测试样本输入模型后, 将会激活后门实施攻击。对模型注入后门, 训练中毒模型参数 θ^* 的过程如下:

$$\theta^* = \min_r \mathbb{E}_{x \in \mathcal{X}} [\ell(x + r, t; \theta)] \quad (3)$$

其中, θ 表示模型的训练参数, $x \in \mathcal{X}$, 为正常样本集中的一张样本; $\ell(\cdot)$ 为损失函数, r 表示触发器。训练时, 对于带有触发器的样本, 模型将学习触发器的特征, 将其与中毒目标类 t 进行关联。而对于良性样本, 模型依旧学习其原有的特征。在测试阶段, 当携带有触发器 r 的样本输入到模型中, 将会刺激模型注入的后门, 使模型的预测输出均变成中毒目标类 t , 触发攻击。而对于不带触发器的正常样本, 模型正常分类。

2.2 可靠性测试

为了防止深度模型在应用时出现由于模型训练、对抗或中毒引起的不确定性输入、偏见操控和

隐私泄露而产生的风险,在部署前需要对其进行可靠性测试,通过生成测试样本以评估模型风险,发现潜在缺陷,使其在工作时安全可靠。

与开发人员直接指定系统逻辑的传统软件不同,深度学习模型自动从数据中学习特征规律,这些规律大多不为开发人员所知。因此,深度学习的可靠性测试,依托神经网络所学习到的规律,查找触发错误行为的漏洞输入,并提供如何使用这些输入修复错误行为的初步依据。

2.2.1 可靠性测试的目标

测试目标是指在深度模型测试中要测试的内容,通常与训练后深度模型的行为相关。应用时可靠的深度学习模型需要满足分类准确、风险鲁棒、决策公平、隐私安全的基本条件。因此,我们将测试目标分为模型任务性能、安全性、公平性和隐私性。

任务性能。模型任务性能可以分为分类准确率和训练程度。准确率衡量深度模型能否正确判断输入的样本。分类准确率定义如下:

$$E(h) = Pr_{x \sim \mathcal{D}}[h(x) = c(x)] \quad (4)$$

其中, x 为数据集 \mathcal{D} 中的单条数据, $h(x)$ 为模型对于 x 的预测类标, $c(x)$ 为 x 的真实类标, Pr 表示概率。分类准确率 $E(h)$ 为预测类标与真实类标相同的概率。分类准确率越高,模型对于未知输入就越容易做出正确的判断。

模型的训练程度评估检测模型和数据之间的匹配程度,其定义如下:

$$f = |R(\mathcal{D}_1, A) - R(\mathcal{D}_2, A)| \quad (5)$$

其中, \mathcal{D}_1 和 \mathcal{D}_2 为具有相同分布的数据, A 为待测模型, $R(\cdot)$ 衡量模型和数据的匹配程度。匹配程度越高,模型训练程度越高,模型性能越好。较低的训练程度通常与过拟合或欠拟合有关。当模型对于数据来说过于复杂且训练数据不足时,容易发生拟合现象。

安全性。模型的安全性衡量了模型抵御内生脆弱性和由外部攻击所引起的风险的能力,其定义如下:

$$r = E(S) - E(S') \quad (6)$$

其中, S 表示深度学习系统,包括数据、模型参数等, S' 表示扰动后的数据或模型。 $E(\cdot)$ 表示模型分类结果。安全性较低的模型往往在训练阶段容易受到中毒攻击的影响,在推理阶段易受对抗攻击而出现误判断。

公平性。由于训练数据和模型结构设计上存在偏见,深度模型的预测结果会在敏感属性(例如性别、种族等)方面存在偏见的现象。公平性测试可以被分为群体公平和个体公平。决策公平的模型在面

对具有相同敏感属性的群体或个体时,会输出相同的判断。群体公平定义如下:

$$Pr(h(x_i) = 1 | x_i \in G_1) = Pr(h(x_j) = 1 | x_j \in G_2) \quad (7)$$

其中, G_1 和 G_2 是拥有共同敏感属性的群体, x_i 和 x_j 是分别来自 G_1 和 G_2 的样本, $h(\cdot)$ 为模型预测类标。

个体公平定义如下:

$$Pr(h(x_i) = a | x_i \in X) = Pr(h(x_j) = a | x_j \in X) \quad (8)$$

其中, X 表示具有相同敏感属性 a 的样本集。模型的公平性衡量了模型输出结果的公平程度,公平性越高,则模型输出结果存在的偏见越少,可靠性越高。

隐私性。模型的隐私性指的是模型对于私密数据信息的保护能力,常用差分隐私进行定义:

$$Pr(h(\mathcal{D}_1) \in Y) \leq e^\epsilon Pr(h(\mathcal{D}_2) \in Y) \quad (9)$$

其中, \mathcal{D}_1 和 \mathcal{D}_2 为同一训练集中的子集,仅有一张样本存在差异。 Y 为模型 $h(\cdot)$ 的输出集合, ϵ 为非常小的常数。隐私保护能力强的模型在接受多次查询时的输出概率保持一致,使攻击者难以推断训练集中的隐私信息。深度模型的隐私性越高,在应用时就更为可靠。

2.2.2 可靠性测试的过程

目前面向深度模型的测试研究主要集中在离线测试,其工作流程如图6所示。由测试种子生成或选取测试样本作为测试输入,以模型输出计算的测试指标为指导,进一步生成测试样本。在深度模型中测试这些样本,获得最终的测试报告。测试样本的生成过程可以采取多种方式,包括模糊测试和符号执行。

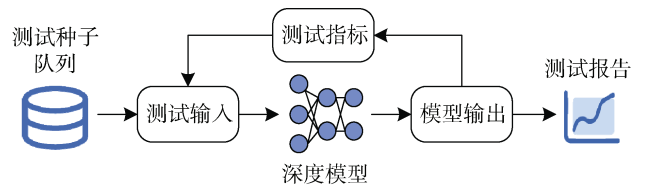


图6 测试流程图

Figure 6 The pipeline of testing procedure

模糊测试通过多种策略随机生成大量测试样本来进行多次测试,以观察模型是否出现误分类,由此发现模型中可能存在的漏洞^[17]。这些策略可以基于变异、搜索和组合测试。基于变异的方法对现有的测试样本进行变异以生成新的测试样本^[18]。组合测试^[19]是一种平衡测试探索和缺陷检测能力的有效测试技术,在获得期望的覆盖率的同时最小化测试集的大小。

符号执行通过分析程序,找到能使待测模型出

现错误的符号化输入^[20]。自动符号执行, 又称 concolic 测试^[21], 将程序执行和符号分析相结合, 通过自动生成测试输入实现高覆盖率。

3 任务性能测试

模型的任务性能主要包括准确率和训练程度, 涉及深度模型的基本功能准确性。分类准确率低、训练不足或过拟合的模型往往不能部署到实际应用中。本节将对准确率和训练程度两个方面的测试方法进行介绍。

3.1 模型准确率测试

准确率是模型基础性能之一, 它衡量了模型是否能对输入做出正确的判断。准确率有很多种衡量指标, 包括准确率、精确率、召回率、接受者操作特征(Receiver Operating Characteristic, ROC)曲线和曲线下面积(Area Under Curve, AUC)。

3.1.1 测试指标

模型准确率的衡量指标包括准确率、精确率、召回率、ROC 曲线和 AUC。

基于样本预测值和真实值是否相符, 可得到以下 4 种结果:

(1) 真阳性(True Positive, TP), 即样本预测值与真实值相符且均为正。

(2) 假阳性(False Positive, FP), 即样本预测值为正而真实值为负。

(3) 假阴性(False Negative, FN), 即样本预测值为负而真实值为正。

(4) 真阴性(True Negative, TN), 即样本预测值与真实值相符且均为负, 即分类准确率。

根据这四种情况, 可以计算以下准确率指标。

准确率的定义是预测正确的结果占总样本的百分比, 其公式如下: $Acc = \frac{TP + TN}{TP + FP + TN + FN}$ 。虽

然准确率可以在总体上判断正确率, 但是在样本不平衡的情况下, 并不能作为很好的指标来衡量结果。

精确率又叫查准率, 其计算公式为: $Precision = \frac{TP}{TP + FP}$, 它是针对预测结果而言的, 表示在所有预测为正的样本中实际为正的样本的概率。精确率代表对正样本结果中的预测准确程度, 而准确率则代表整体的预测准确程度, 既包括正样本, 也包括负样本。

召回率也称为真阳性率, 计算如下: $Recall = \frac{TP}{TP + FN}$ 。它是针对原样本而言的, 含义是在实际为正的样本中预测为正样本的概率, 代表分类器预测

的正类中实际正实例占有所有正实例的比例。

ROC 曲线可以用于评价一个分类器在不同阈值下的表现情况。其中, 每个点的横坐标是假阳性率, 纵坐标是真阳性率, 描绘了分类器在 TP 和 FP 间的平衡。其中假阳性率计算如下: $FPR = \frac{FP}{FP + TN}$, 代

表分类器预测的正类中实际负实例占有所有负实例的比例。ROC 曲线越接近左上角, 该分类器的性能越好。

AUC定量度量了 ROC 曲线, 计算了其下方的面积, 范围为 0~1。AUC 越大, 说明模型分类性能越好。

3.1.2 测试方法

上节中介绍的各种准确率指标已在学术界被广泛采用, 以衡量模型的性能。但在数据不平衡时, 它们无法全面衡量分类的正确性。例如, 准确性不能反映假阳性和假阴性, 精确率和召回率可能会误导结果。

为了更好地选择测试指标, 并发现模型分类正确性问题, Japkowicz 等人^[22]对各个准确率指标进行了分析, 指出了各自的缺点, 并提醒学者们谨慎选择准确率指标。Chen 等人^[23]在评估模型分类的正确性时, 研究了训练数据和测试数据的可变性。他们推导出了估计性能方差的解析表达式, 并提供了用高效计算算法实现的开源软件。他们还研究了比较 AUC 时不同统计方法的性能^[24], 发现 F-test 的性能优于其他方法。Qin 等人^[25]提出了基于程序合成的方法 SynEva, 从训练数据生成镜像程序, 然后使用该镜像程序的行为作为准确性的参考标准。镜像程序会具有和测试数据类似的行为。

3.2 训练程度测试

模型的训练程度评估了模型和数据的匹配程度, 训练不足或过度都可能导致模型性能不佳, 影响分类准确性。在训练数据不足, 或模型对于训练数据来说太复杂的情况下容易发生过拟合。

交叉验证传统上被认为是一种检测过拟合的有效方法。针对深度模型, Zhang 等人^[26]引入了扰动模型验证(Perturbed Model Validation, PMV), 判断模型是否训练完全, 来帮助模型选择。他们将噪声注入到训练数据中, 针对受干扰的数据对模型进行再训练, 然后利用训练精度下降率来评估模型和数据的匹配程度。结果表明, PMV 比交叉验证更准确、更稳定地选择模型, 并能有效地检测过拟合和欠拟合。Werpachowski 等人^[27]提出了一种通过从测试数据生成对抗样本的过拟合检测方法。如果对抗样本的重新加权误差估计与原始测试集的误差估计差距悬殊, 则模型存在过拟合的问题。Chatterjee 等人^[28]提出了

一系列称为反事实模拟(Counterfactual Simulation, CFS)方法, 该方法只需要通过逻辑电路表示来分离具有不同级别过拟合的模型, 而不用访问任何模型的高级结构。

对于过拟合产生的原因, Gossmann 等人^[29]在医学测试数据集上进行多次实验, 发现重复使用相同的测试数据会造成模型过拟合。Ma 等人^[30]对训练集进行重采样来缓解过拟合问题。

由于模型容量和输入数据的分布通常是未知的, 因此, 模型训练程度的测试具有挑战性。目前, 还缺乏完善的针对模型欠拟合的测试方法。此外, 模型训练程度和安全性之间的平衡关系也值得探究。

4 安全性测试

随着深度模型在越来越多的场景中得到应用, 其安全性受到了广泛关注。深度模型由于内生脆弱性, 在推理阶段容易受到不确定性输入而出现误判断, 在训练阶段则容易由于污染的数据而留下中毒后门。对此, 现有按照威胁的不同阶段, 面向模型安全性的测试方法可以分为两类: 面向推理阶段和训练阶段安全性的测试方法。前者主要针对

推理阶段的不确定输入, 后者对训练阶段隐藏的后门进行检测。此外, 本节将介绍测试样本的选取方法, 对大量测试样本进行选择 and 优先级排序, 实现高效测试。

4.1 推理阶段安全性测试

推理阶段的安全性测试主要针对输入的不确定性。根据测试样本生成的指导指标, 它们可以分为基于覆盖率的方法和基于变异的测试方法。此外, 基于修复的方法生成样本对模型进行调试和修复, 提高其鲁棒性和分类精度。

4.1.1 基于覆盖率的测试方法

传统软件测试中, 测试覆盖率是衡量测试充分性的一个重要指标, 能够帮助客观认识软件质量, 提高测试效果。受此启发, 深度模型测试借鉴了覆盖率的概念来定量衡量测试的充分性, 并指导性测试样本的生成。覆盖率指标可以分为神经元覆盖率、多粒度神经元覆盖率、改进条件/判定范围(Modified Condition/Decision Coverage, MC/DC)覆盖率和状态覆盖率, 具体方法总结如表 1。这些方法认为, 经过更系统测试(即具有更高覆盖率)的深度模型, 在面对不确定性输入时更安全可靠。

表 1 基于覆盖率的测试方法总结
Table 1 Summary of coverage-based testing methods

指标类型	测试方法	覆盖率指标	样本生成方法	适用模型	应用领域
神经元覆盖率	DeepXplore ^[16]	NC	差分	CNN	自动驾驶
	DLFuzz ^[31]	NC	差分模糊测试	CNN	图像
	TensorFuzz ^[34]	NC	变异模糊测试	CNN、RNN	图像、语音
	DeepTest ^[35]	NC	差分贪婪搜索	CNN	自动驾驶
多粒度 神经元覆盖率	DeepGauge ^[36]	NC、KMNC、NBC、SNAC、TKNC、TKNP	差分	CNN	图像
	DeepHunter ^[37]	NC、KMNC、NBC、SNAC、TKNC、TKNP	变异模糊测试	CNN	图像
	ADAPT ^[38]	NC、TKNC	差分	CNN	图像
	DeepCT ^[19]	SparseCov、DenseCov	组合测试	CNN	图像
MC/DC 覆盖率	DeepCover ^[39]	SSC、VSC、SVC、VVC	符号执行	CNN	图像
	DeepConcolic ^[21]	NC、SSC、NBC	Concolic	CNN	图像
状态覆盖率	DeepCruiser ^[40]	BSC、k-SBC、BTC、ISC、WIC	变异	RNN	语音、文本
	DeepStellar ^[41]	BSC、WSC、n-SBC、BTC、WTC	变异	RNN	图像、语音
	testRNN ^[42]	NC、BC、SC、TC	变异	RNN	图像、语音

神经元覆盖率。Pei 等人^[16]提出了首个用于系统地测试深度模型的白盒框架 DeepXplore(如图 7 所示), 并引入了神经元覆盖率(neuron coverage, NC)指标来评估由测试输入执行的模型的各个部分。DeepXplore 在最先进的深度学习模型中有效地发现了数千个不正确的极端情况行为, 例如, 自动驾驶汽车撞到护栏,

恶意软件被识别为良性软件。基于此, Guo 等人^[31]提出了首个差分模糊测试框架 DLFuzz, 不断对输入进行细微的变异, 以最大化神经元覆盖率以及原始输入和变异输入之间的预测差异, 用于指导模型漏洞检测, 其框架如图 8 所示。在 MNIST^[32]和 ImageNet^[33]数据集上, 与 DeepXplore 相比, DLFuzz 消耗更少的时间

间, 获得了更高的覆盖率。Odena 等人^[34]开发了覆盖率引导模糊测试方法 TensorFuzz, 以发现仅在稀有输入中发生的错误。输入的随机突变由覆盖度量引导, 以达到指定约束的目标。Tian 等人^[35]设计了 DeepTest, 用于自动检测深度模型驱动的车辆的行为。通过生成最大化神经元覆盖率的测试输入来系统地探索深度模型的不同部分, 在不同的现实驾驶条件下(例如, 模糊、下雨、雾等)发现了数千种错误行为。

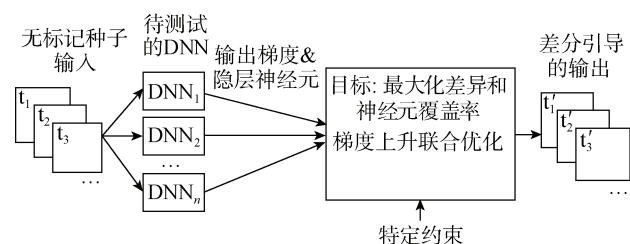


图 7 DeepXplore 的总体框架^[16]

Figure 7 The framework of DeepXplore^[16]

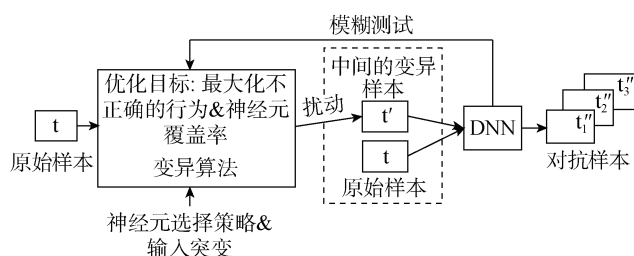


图 8 DLFuzz 的总体框架^[31]

Figure 8 The framework of DLFuzz^[31]

多粒度神经元覆盖率。Ma 等人^[36]将神经元覆盖率进行了更细粒度地划分, 提出了一套基于多层次、多粒度覆盖的测试准则 DeepGauge。其中包括了强神经元激活覆盖率(strong neuron activation coverage, SNAC), k 节神经元覆盖率(k -multisection neuron coverage, KMNC), 前 k 个活跃神经元覆盖率(top- k neuron coverage, TKNC), 前 k 个活跃神经元模式(top- k neuron coverage patterns, TKNP)和神经元边界覆盖率(neuron boundary coverage, NBC)。利用这五个指标, Xie 等人^[37]提出了一种覆盖引导的模糊测试框架 DeepHunter, 达到了更高的覆盖率并发现了更多数目和种类的模式缺陷。基于 NC 和 TKNC, Lee 等人^[38]提出了 ADAPT, 通过自适应的神经元选取策略实现更系统和有效的深度模型测试。此外, Ma 等人^[19]引入了将组合测试的概念, 提出了 DeepCT 来平衡缺陷检测能力和测试样本的数量。同时, 他们提出了组合测试指标, 包括 t 路组合稀疏覆盖率(t -way combination sparse coverage, SparseCov)和 t 路组合密集覆盖率(t -way combination

dense coverage, DenseCov), 通过相对较少的测试来实现合理的缺陷检测能力。

MC/DC 覆盖率。传统软件测试中的 MC/DC 覆盖要求每个条件都对最终结果起独立作用。Sun 等人^[39]把这个概念应用到神经网络中, 把上一层的所有神经元看作一个分支条件 bool 表达式中的各个子条件, 本层的某个神经元看作是结果, 由此为测试 DNN 提出了新思路。基于此, 他们提出了能够直接应用于 DNN 模型的四个测试覆盖率标准和基于线性规划的测试样本生成算法, 在小型神经网络中表现出较高的缺陷检测能力。此外, Sun 等人^[21]开发了首个面向深度模型的 concolic 测试方法 DeepConcolic(如图 9 所示), 将具体执行和符号分析进行结合, 实现了更高的覆盖率。

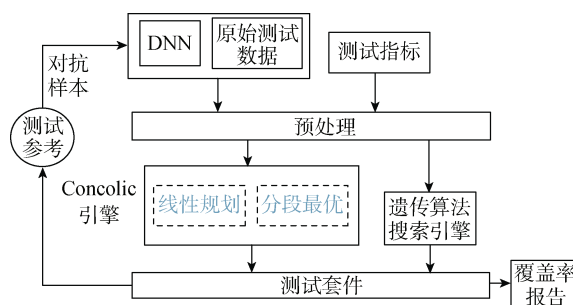
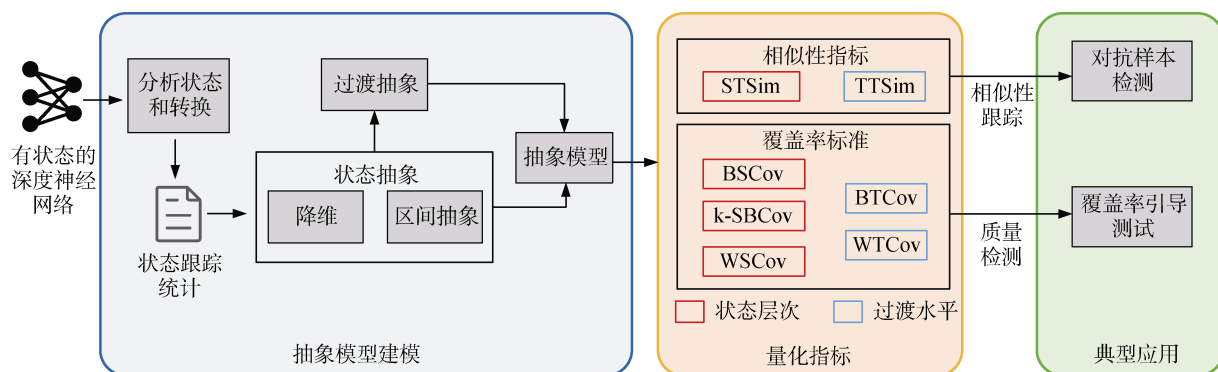


图 9 DeepConcolic 的总体框架^[21]

Figure 9 The framework of DeepConcolic^[21]

状态覆盖率。Du 等人^[40]将有状态的 RNN 建模为一个抽象的状态转换系统, 并定义了一组专门用于有状态深度模型的测试覆盖标准。此外, 他们提出了一个自动化测试框架 DeepCruiser, 它可以系统地大规模生成测试样本, 以通过覆盖率指导发现有状态深度模型的缺陷, 在语音和文本数据集上证明了测试质量和可靠性方面的有效性。在此基础上, 他们设计了状态和转移轨迹的相似性指标和五个状态覆盖率准则, 并提出了 DeepStellar^[41], 对 RNN 进行引导测试(如图 10 所示)。实验证明, 他们提出的相似性指标能有效检测出对抗性输入。Huang 等人^[42]提出了覆盖率引导的测试工具 testRNN, 并提出了四个状态覆盖率指标, 用于验证 LSTM。该工具基于变异生成测试样本, 使用三个新的 LSTM 结构测试覆盖度量来评估网络的鲁棒性。此外, 它还能它能够帮助模型设计者完成 LSTM 层的内部数据流处理。

覆盖率方法的局限性。目前已提出的覆盖率指标基于以下假说: 神经元覆盖率与对抗性输入的生成和测试方法的错误揭示能力相关。但新的研究对此提出了质疑。

图 10 DeepStellar 的总体框架^[41]Figure 10 The framework of DeepStellar^[41]

目前提出的大多数覆盖率指标都基于深度模型的结构,但神经网络与程序软件之间存在根本差异,这导致了当前覆盖率指标的局限性。Li 等人^[43]发现从高覆盖率测试中推测的故障检测“能力”,更有可能是由于面向对抗性输入的搜索,而不是真正的“高”覆盖率。他们对自然输入的初步实验发现,测试集中错误分类的输入数量与其结构覆盖率之间没有强相关性。由于深度模型的黑盒性质,尚不清楚这些指标如何与系统的决策逻辑直接相关。

Harel-Canada 等人^[44]设计了一个新的多样性促进正则器,在缺陷检测、输入真实性和公正性三个方面对覆盖率指导的测试样本进行评估,发现神经元覆盖率增加反而降低了测试样本的有效性,即减少了检测到的缺陷,产生了更少的自然输入和有偏的预测结果。

通常认为,用覆盖率更高的测试样本对模型进行重训练,可以提高模型的鲁棒性。对此,Dong 等人^[45]对 100 个深度模型和 25 个指标进行了实验,结果说明覆盖率和模型鲁棒性之间的相关性有限。因此,高覆盖率对提高模型的鲁棒性没有意义。

因此,基于高覆盖率生成测试样本可能是片面的,在深度模型测试中衡量测试有效性还有很大的改进空间。

4.1.2 基于变异的测试方法

在传统的软件测试中,变异测试通过注入故障来评估测试套件的故障检测能力^[46]。检测到的故障与所有注入故障的比率称为变异分数。

在深度模型的测试中,数据和模型结构也在一定程度上决定了模型的行为。Ma 等人^[17]提出了专门用于深度模型的变异测试框架 DeepMutation,不观察模型的运行时内部行为,通过定义一组源级或模型级变异算子,从源级(训练数据和训练程序)或模型级(无需训练直接注入)注入故障用来以评估测试数

据质量。他们在小尺寸数据集上证明了方法的有效性。在此基础上,Hu 等人^[47]提出了 DeepMutation++,对模型质量进行评估,尤其是 RNN。它不仅能够针对整个输入静态分析模型的鲁棒性,而且还能在模型运行时分析识别顺序输入(例如音频输入)的脆弱部分。在此基础上,Humbatova 等人^[48]定义了 35 个变异算子,并提出了基于真实故障的源级预训练变异工具 DeepCrime,取得了更优异的性能。Shen 等人^[49]提出了 MuNN,利用五个变异算子评估了 MNIST 数据集上的变异属性,并指出需要域特定的变异算子来增强变异分析。Riccio 等人^[50]提出了一种自动生成新测试输入的方法 DeepMetis,利用基于搜索的输入生成策略生成测试输入来提高变异分数,模拟尚未检测到的故障的发生。

与覆盖率标准相比,基于变异测试的标准与模型的决策边界更直接相关。靠近模型决策边界的输入数据可以更容易地检测模型与其变异体之间不一致。

4.1.3 基于修复的测试方法

与软件漏洞不同,深度模型的缺陷不能通过直接修改模型来轻松修复。受到软件调试的启发,基于修复的测试方法通过选择合适的输入对模型进行调试,提高其精度或鲁棒性。

Ma 等人^[29]提出了一种新的模型调试技术 MODE,评估每一层以识别有缺陷的神经元,并进一步生成用于再训练的修复补丁。该方法可以有效地修复模型错误,避免了新漏洞的产生,但是修复工作可能仅限于那些具有复杂特征的数据集。Zhang 等人^[51]设计了一种权重自适应方法 Apricot 来迭代地修复模型(如图 11 所示)。他们用原始训练集的不同子集训练缩减模型,以此提供权重大小和方向的调整依据。Apricot 既不需要额外的样本,也不需要额外训练神经网络架构,因此更为通用。在此基础上,他们还提出

了超启发式模型修复方法 Plum^[52], 通过制定一系列优先级方案来对修复策略进行排序, 以寻求每个策略在验证和测试数据集上的模型性能之间的平衡。

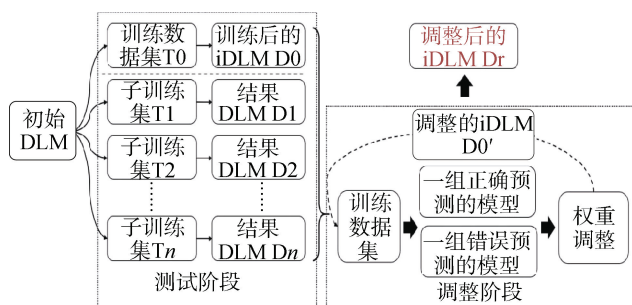


图 11 Apricot 的总体框架^[51]

Figure 11 The framework of Apricot^[51]

从提高模型鲁棒性的角度出发, Borkar 等人^[53]提出了 DeepCorrect, 修复深度学习模型中的卷积滤波器, 以增强对图像失真的鲁棒性。通过对各个滤波

器计算校正优先级, 对最易受噪声影响的滤波器进行重新训练, 同时保持网络中其余预训练滤波器输出不变。但该方法忽略了其他类型层中的其他潜在缺陷。Eniser 等人^[54]提出了 DeepFault(如图 12 所示), 观察了模型中的每个神经元, 根据模型的正确或错误行为调查神经元是否被异常激活。对于那些可疑值较高的神经元, 他们合成由这些神经元的梯度并通过合成输入进行再训练来纠正模型的行为。Wang 等人^[55]提出一种称为鲁棒性导向测试(Robustness-Oriented Testing, RobOT)的新型测试框架, 提出了零阶损失(Zero-Order Loss, ZOL)和一阶损失(First-Order Loss, FOL)指标, 在提高模型鲁棒性方面对测试样本进行定量测量, 使多个基准数据集上的模型对抗性鲁棒性提高了 67.02%。Gao 等人^[56]提出了一种基于突变的模糊测试方法 Sensei 和 Sensei-SA 来增强模型的训练数据, 在 15 个模型中分别使其鲁棒性评价提高了 11.9%和 5.5%。

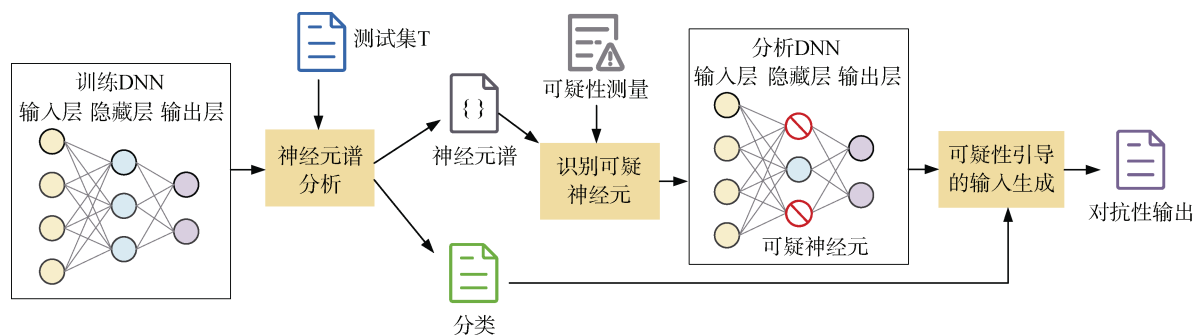


图 12 DeepFault 的工作流^[54]

Figure 12 The workflow of DeepFault^[54]

此外, 针对 RNN 驱动的对话系统, Liu 等人^[57]提出了 DialTest, 在保留标记信息的情况下, 通过同义词替换、回译和单词插入对测试种子进行实际更改, 有效地检测错误并提高系统的鲁棒性。

为了进一步提高模型精度, 也有研究通过数据增强来修复模型。Yu 等人^[58]提出了一种基于样式转移的数据增强的修复方法 DeepRepair, 用于在操作环境中修复深度模型。他们也进一步提出了基于聚类的故障数据生成, 以实现更有效的风格引导数据增强。在医学诊断方面, Hou 等人^[59]提出 TauMed, 它基于医疗数据集上的一系列变异规则和语义实现增强技术, 生成足够数量且高质量的图像, 以提高模型分类精度。

通过增加样本或修改模型, 基于修复的测试方法提高了模型的分类精度或鲁棒性, 其总结如表 2 所示。

4.2 训练阶段安全性测试

当输入的样本存在特定的触发器时, 在训练阶段被植入中毒后门的模型会输出错误的预测结果。目前对于中毒后门的检测方法可以分为离线和在线检测, 它们通过不同的方式观察模型的异常行为, 确定模型是否存在中毒后门。

4.2.1 离线检测

离线检测在离线状态下检测模型中的中毒后门, 往往需要比较高的计算成本和专业的模型知识。

Wang 等人^[60]提出了 NeuralCleanse, 他们认为带有后门的模型所对应的触发器, 要比利用正常模型生成的触发器要小得多。通过观察模型的神经元激活向量中潜在表示的异常情况来检测它在部署之前是否被中毒。在此基础上, Guo 等人^[61]提出了 TABOR, 利用正则化解决优化问题, 取得了检测结果的提升。针对 NeuralCleanse 在多类数据集上计算成本很高的

表 2 基于修复的测试方法总结

Table 2 Summary of debugging based testing methods

方法	修复手段	核心思想	修复目标	适用模型	应用领域
MODE ^[29]	增加样本	缺陷神经元	分类精度	CNN	图像
Apricot ^[51]	修改模型	权重自适应	分类精度	CNN	图像
Plum ^[52]	排序修复策略	评估修复策略并确定优先级	分类精度	CNN	图像
DeepCorrect ^[52]	修改模型	校正易受影响的卷积滤波器	分类精度	CNN	图像、目标检测、场景识别
DeepFault ^[54]	增加样本	异常激活神经元	分类精度	CNN	图像
DeepRepair ^[58]	增加样本	样式引导的数据增强, 基于聚类的样本生成	分类精度	CNN、RNN	图像
TauMed ^[59]	增加样本	基于变异和语义的数据增强	分类精度	CNN	医疗图像
DialTest ^[57]	增加样本	基于 Gini 系数的模糊测试	鲁棒性	RNN	自然语言理解
RobOT ^[55]	增加样本	FOL、ZOL 定量衡量测试样本	鲁棒性	CNN	图像
Sensei、Sensei-SA ^[56]	增加样本	基于突变的模糊测试	鲁棒性	CNN	图像

问题, Harikumar 等人^[62]提出了可拓展的木马扫描程序(Scalable Trojan Scanner, STS), 以识别触发器的方式对其进行反转, 由此将搜索过程与类别数目独立开来。Chen 等人^[63]提出了 DeepInspect, 使用条件生成模型来学习潜在触发器的概率分布。该生成模型将用于生成反向触发器, 统计评估其扰动水平, 以构建后门异常检测。基于分解和统计分析, Tang 等人^[64]利用类别的全局信息提出了 SCAN, 以检测已知或未知的中毒攻击。

不同于以上方法专门用于图像分类领域, Xu 等人^[65]设计了一种适用于语音和文本场景下的后门检测方法 MNTD, 通过训练元分类器以预测模型是否安全。

4.2.2 在线检测

与离线检测不同, 在线检测可以在模型运行时

对其行为进行检测。

Liu 等人^[66]提出了人工脑刺激(Artificial Brain Stimulation, ABS), 通过刺激的方法找到中毒模型中的受损的神经元, 在其指导下进一步反转触发器, 以确定模型是否被中毒。ABS 与触发器尺寸无关, 能检测对模型特征空间的中毒后门, 但不能处理像素空间的后门攻击。此外, 他们也提出了 EX-RAY^[67], 基于对称特征差分方法来区分自然特征和中毒特征。

4.2.3 小结

面向训练阶段安全性的测试方法优缺点总结如表 3 所示。这些方法主要为计算机视觉领域的分类任务设计, 目前缺乏在语音和文本等不同领域的通用对策。此外, 在训练阶段针对 RNN 等序列模型的测试也相当重要, 却少有研究涉及。

表 3 训练阶段安全性测试方法优缺点总结

Table 3 Summary of testing methods in the training stage

测试方法	离线/在线	优点	缺点
NeuralCleanse ^[60]	离线	首个通用的后门检测和防御方法	多类数据集计算成本高, 难以处理大尺寸的触发器
TABOR ^[61]	离线	更高的检测效率	对像素空间后门攻击的性能没有得到证实
STS ^[62]	离线	多类数据集上节约时间	难以处理大尺寸的触发器
DeepInspect ^[63]	离线	在复杂数据集上计算速度快	检测效率低
SCAN ^[64]	离线	检测已知或自适应的数据污染攻击	依赖于触发本来识别中毒类
MNTD ^[65]	离线	与攻击策略无关, 适用于文本和语音场景	需要大量计算资源, 不太实用
ABS ^[66]	在线	仅需要少量输入样本实现高检测率, 可以反转大尺寸触发器	不能处理像素空间的后门攻击和复杂的特征攻击
EX-RAY ^[67]	在线	消除了大部分误报率, 检测精度提高	难以反转像素空间后门攻击的触发器

4.3 测试样本选取方法

生成面向深度模型的测试样本往往需要涵盖非常大的输入空间, 而这些测试样本需要花费昂贵的人

力成本来进行标记, 显著影响可执行的测试的数量和质量。因此, 以有意义的方式优先选择测试模型的输入数据, 可以降低标记成本, 大大提高测试效率。

Feng 等人^[68]基于深度模型的统计视角, 首先提出了测试样本优先级排序技术 DeepGini, 可以快速识别可能导致模型误分类的测试样本。DeepGini 在优先级排序的有效性和运行效率方面优于基于覆盖率的方法。Byun 等人^[69]根据 softmax 置信度、贝叶斯不确定性和输入意外程度对测试样本进行优先级排序, 在图像分类和图像回归的数据集上取得了 70% 以上的平均缺陷检测率。

Zhang 等人^[70]提出了一种基于噪声敏感性分析的测试优先排序技术 NSATP, 通过噪声敏感性来挑选样本。与他们不同, Zhang 等人^[71]观察了神经元的激活模式, 根据训练集中获得的神经元的激活模式和从特定输入中收集的激活神经元来计算测试样本的优先级。结果表明, 具有较高优先级的测试样本更容易被误分类。此外, 对相同数据集中的模型进行优先排序的测试样本, 也能使其他具有相似结构的模型误分类。

Ma 等人^[72]基于模型不确定性的概念, 对一组测试选择度量进行了深入的经验比较。他们优先选择了具有更高不确定性的样本, 用其对模型进行训练, 以提高分类精度。Wang 等人^[73]优先考虑那些通过许多变异而产生不同预测结果的测试输入, 认为它们更有可能揭示出模型的缺陷。由此他们提出了 PRIMA(如图 13 所示), 对变异结果进行智能组合, 从而对测试样本进行优先排序。在现实自动驾驶场景中的实验结果证明了 PRIMA 在排序效果方面的实用性。

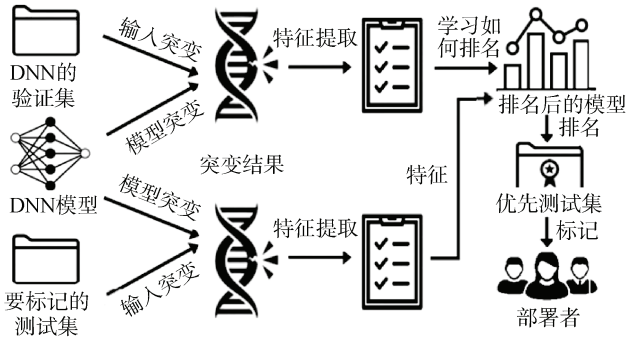


图 13 PRIMA 的总体框架^[73]
Figure 13 The framework of PRIMA^[73]

Kim 等人^[74]将输入的意外程度, 即样本在输入和训练数据之间的行为差异, 作为衡量样本对于深度模型测试充分性的指标。他们认为, 与训练数据相比, 一个理想的测试输入, 应是充分的, 但不应具有很高的意外程度。

基于决策边界的概念, Shen 等人^[75]提出了多边界聚类 and 优先级方法(Multiple-Boundary Clustering and Prioritization, MCP), 将测试样本聚类到模型多个边界区域, 并根据优先级从所有边界区域均匀选择样本。在有效性方面, 经过 MCP 评估再训练的深度学习模型性能得到显著提高。

为了选择可以区分多个模型的有效测试样本, Meng 等人^[76]提出了基于样本区分的选择方法(Sample Discrimination based Selection, SDS), 利用这些样本指示模型的性能。实验证明, 具有更高优先级的样本更有助于模型性能的排序。Guerriero 等人^[77]在数据

表 4 面向测试样本的选取方法总结
Table 4 Summary of selecting testing examples

方法	操作	核心思想	应用场景	局限
DeepGini ^[68]	排序	优先选取输出置信度平均的样本	图像	无法对分类置信度差异悬殊的样本进行选择
Byun et al. ^[69]	排序	Softmax 置信度、贝叶斯不确定性、输入意外程度 LSA	图像	计算成本高
NSATP ^[70]	排序	噪声灵敏度	图像	未提供有关鲁棒增强的指导
Zhang et al. ^[71]	排序	神经元激活模式和频率	图像	需要获得训练集的先验知识; 神经元经验性频率需要一定量的统计; 模型不能存在过拟合与欠拟合现象
Ma et al. ^[72]	排序	对于特定输入的置信度	图像	无法选取对抗样本
PRIMA ^[73]	排序	对突变进行智能组合	图像、文本、自动驾驶	需要一定量的训练集的先验知识; 计算过程空间复杂度, 不仅需要存储特征值, 还需要存储特征对应样本的梯度统计值的索引
SADL ^[74]	选择	计算样本对于训练集的意外程度 DSA	图像、自动驾驶	需要获得训练集的先验知识; 模型不能存在过拟合与欠拟合现象
MCP ^[75]	选择	多决策边界聚类	图像	极端情况的样本可能远离边界, 可能无法找全
SDS ^[76]	选择	对模型预测类标进行投票	图像	准确率依赖多模型预测投票频率
DeepEST ^[77]	选择	非比例选择容易分错的样本	图像	评估模型未考虑鲁棒性依据

集中采用非比例选择查找容易分类错误的测试样本, 提出了一种测试选择技术 DeepEST。与传统抽样相比, 这种方法可以提供更小的方差, 从而使样本选择变得更稳定。

面向测试样本的选取方法总结如表 4 所示。如何高效地对测试样本进行排序, 并以此提高模型的安全性, 成为当前测试研究中的热门话题和前沿方向。

5 公平性和隐私性测试

公平性和隐私性虽不直接影响模型的准确率, 但一定程度上影响了模型的分类行为。本节中我们首先介绍了适用于深度模型的公平性测试方法, 针对模型隐私的测试方法也在后续部分进行阐述。

5.1 公平性测试

公平的研究侧重于发现、衡量、理解和应对观察到的不同群体或个人在表现上的差异。这种差异与偏见有关, 它可能会冒犯甚至伤害用户, 导致种种社会问题。公平性测试旨在发现和减少深度模型中的偏见, 以提升模型在相关领域中应用时的公平性和可靠性。总体而言, 根据不同的测试目标, 公平性测试可以分为个体公平测试和群体公平测试。

5.1.1 个体公平测试

具有个体公平性的模型应该在相似个体之间给出相似的预测结果。个体公平性测试方法关注模型对于不同个体之间的公平性。

Udeshi 等人^[78]提出了 Aequis, 生成测试样本以发现偏见性输入, 以理解个体公平。Aequis 首先对输入空间进行随机采样以发现偏见样本, 然后搜索这些输入的邻域以找到更多偏见。除了检测公平性错误外, Aequis 还重新训练了机器学习模型, 并减少这些模型决策中的偏见。然而, Aequis 对所有输入使用全局采样分布, 这导致它只能在狭窄的输入空间中搜索, 容易陷入局部最优。

Agarwal 等人^[79]设计了一种新的个体公平测试方法 SymbGen, 它将符号执行与本地解释相结合, 以生成有效的测试样本。尽管 SymbGen 是为人工智能模型设计的, 但它仍然使用了决策树。上述方法主要处理传统的机器学习模型, 不能直接应用于处理深度模型。

Zhang 等人^[80]提出了专门针对深度模型公平性的测试方法 ADF, 基于梯度计算和聚类来搜索模型输入空间中的个体偏见样本, 其总体框架如图 14 所示。实验证明, 基于梯度的指导, 生成偏见样本的有效性和计算效率得到了极大的提高。基于此, Zhang 等人^[81]基于梯度设计了白盒公平测试框架 EIDIG, 采用先验信息来加速迭代优化的收敛。但是, 它仍然存在梯度消失而导致局部优化的问题。Zheng 等人^[82]将模型决策偏见归因为偏见神经元, 并提出了针对 DNN 公平的测试框架 NeuronFair, 在更短的时间内生成更多样化的测试样本。

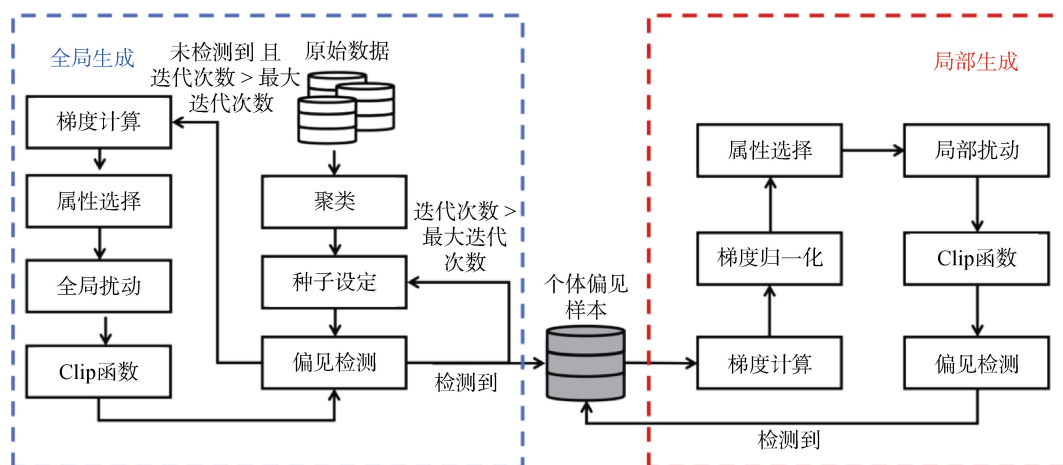


图 14 ADF 的总体框架^[80]

Figure 14 The framework of ADF^[80]

5.1.2 群体公平测试

与个体公平性不同, 群体公平是指模型具有相同的决策概率选择敏感属性的群体。

Galhotra 等人^[83]提出了一种面向群体公平性的软件测试方法 Themis, 它用将公平性分数来衡量软

件的公平性, 并通过计算模型输入空间中个别偏见样本的频率来衡量软件的偏见程度。Sun 等人^[84]提出了一种基于因果关系的神经网络修复技术 CARE, 通过调整给定神经网络的参数, 以修复神经网络获得决策公平性。

5.1.3 小结

针对深度模型的公平性测试处于起步阶段, 如何更全面地定义偏见, 更细粒度地定量描述偏见并以此系统地测试深度模型, 将成为今后公平性测试的主要研究方向。

5.2 隐私性测试

隐私是深度模型保存私人数据信息的能力。在模型部署前需要检查使用敏感数据计算的程序是否保护了用户隐私。

较早提出的方法主要讨论如何确保模型的隐私性。Bichsel 等人^[85]提出了 DP-Finder, 引入了一种有效且精确的采样方法来估计样本泄露隐私的可能性, 并将搜索任务描述为优化目标, 使他们能系统地搜索到大量泄露隐私的样本。

为了检测违反差分隐私的行为, Ding 等人^[86]为差分隐私算法生成违反样本, 多次运行候选算法, 并使用统计测试来检测违反差分隐私的情况。Zhang 等人^[87]提出了首个用于自动化差分隐私的框架 DPCheck, 无需程序员注释, 就能测试算法中的隐私泄露问题。他们在 2020 年美国人口普查信息保护系统上证明了方法的有效性。基于静态程序分析, Wang 等人^[88]提出了 CheckDP 对模型的差分隐私机制进行评估。该方法可以在 70s 内判断模型是否违反了差分隐私机制, 并给出证明或生成反例。Bichsel 等人^[89]训练分类器对差分隐私进行近似最优攻击, 以发现模型在违反差分隐私方面的情况, 提出了 DP-Sniper。实验证明, 他们的方法在有效性和高效性方面具有很大的提升。Farina 等人^[90]设计了一种关系符号执行方法, 支持概率耦合的推理, 可用于构建差分隐私的证据。

对模型隐私的测试方法正受到越来越多的关注, 如何通过测试使模型具有更高的隐私保护能力, 也是学者们关注的关键问题。

6 深度模型可靠性测试的应用

深度模型已广泛应用于不同领域。本节在三个典型的应用领域介绍这种特定领域的测试方法: 自动驾驶、语音识别和自然语言处理。

6.1 自动驾驶

自动驾驶显示出改革现代交通的巨大潜力, 其安全性备受公众关注。因此, 该领域成为测试的重点方向。

Pei 等人^[16]通过基于梯度的差分测试生成测试样本, 来发现模型中的隐藏缺陷。Tian 等人^[35]用多种图像变换来产生可能在现实摄像头中出现的隐藏噪

声, 以此对自动驾驶场景中的不确定输入进行模型缺陷的检测。Zhang 等人^[91]基于对抗性生成网络^[92]生成测试样本, 提出了一种基于无监督的自动一致性测试框架 DeepRoad, 支持雪天和雨天的天气状况。Zhou 等人^[93]提出了 DeepBillboard, 其流程如图 15 所示。它能针对广告牌生成可打印的对抗性测试样本, 能最大化自动驾驶汽车驾驶时转向角错误的可能性、程度和持续时间, 在不同的视角、距离和照明的驾驶条件下都能工作。Riccio 等人^[94]引入了行为边界的概念, 即让深度学习系统开始误分类的输入, 并设计了 DeepJanus, 以基于搜索的方式为深度学习系统生成测试样本。在自动驾驶汽车车道保持系统上的实验证明了其有效性。

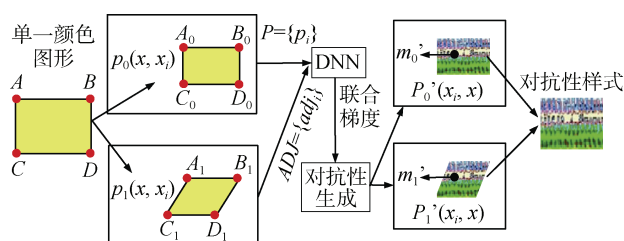


图 15 DeepBillboard 生成对抗性扰动的流程^[93]

Figure 15 The generation of perturbations in DeepBillboard^[93]

自动驾驶场景中深度模型的测试需要大量测试样本, Birchler 等人^[95]对这个场景中的测试样本进行排序, 尽可能早地揭示模型中存在的缺陷。他们利用静态特性区分安全和不安全的场景, 提出了自动驾驶场景的测试样本排序方法 SDC-Prioritizer, 基于遗传算法对模型进行高效测试。

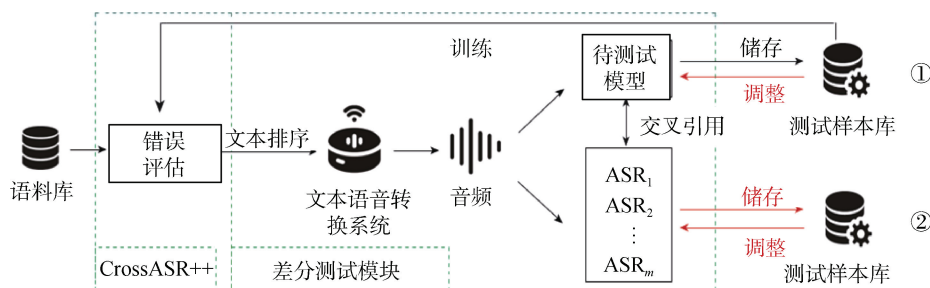
6.2 语音识别

目前也有相关学者在语音场景下对深度模型进行测试, 以证明其安全性和公平性。

Asyrofi 等人^[96]提出了一种利用现有的文本到语音系统来自动生成自动语音识别(Automated Speech Recognition, ASR)系统的测试样本的方法 CrossASR, 如图 16 所示。作为一种黑盒方法, CrossASR 通过失败概率预测, 为任何 ASR 系统生成使其误判可能性最大的测试样本。在此基础上, 他们还提出了 CrossASR++^[97], 它发现更多使模型出错的测试样本。Du 等人提出了 DeepCruiser^[40]和 DeepStellar^[41], 利用状态覆盖率指标生成测试样本, 在实际 ASR 系统的 RNN 模型中取得了更高的覆盖率, 发现了更多系统缺陷。

6.3 自然语言处理

在自然语言处理方面, Sun 等人^[98]在他们的工具

图 16 CrossASR++框架图^[97]Figure 16 The framework of CrossASR++^[97]

“MT4MT”中使用自定义变形关系来测试机器翻译系统的翻译一致性。他们认为输入的改变不应影响翻译后内容的结构,以此评估机器翻译系统的公平性和语义一致性。此外, Sun 等人^[99]将变异测试和变形测试相结合,提出了 TransRepair, 对机器翻译系统谷歌翻译和 Transformer 进行测试,并修复了它们将近 20%的错误。

6.4 小结

当前面向深度学习的测试方法研究主要集中在监督学习上,特别是分类问题。在无监督学习和强化学习相关的应用领域,深度模型的可靠性亟待系统性的测试。

目前研究中的测试任务主要集中在图像分类上。在许多其他领域和任务,如多智能体游戏场景,仍然存在许多开放测试的研究机会。

7 深度模型测试的数据集、在线模型库和常用工具包

近年来用于深度模型测试的模型库和工具包不断涌现,为系统性地测试深度模型提供了便利。本节列举了可用于深度模型测试的数据集,并介绍了现有可测试的在线模型库和常用测试工具包,便于研究者进行后续研究。

7.1 数据集

根据不同的应用场景和任务,数据集可以分为图像分类、语音识别、自然语言处理和模型公平决策这四类。本节中我们将分别对这些数据集进行简单介绍,总结如表 5 所示。

7.1.1 图像分类

MNIST 数据集^[32]是美国国家标准与技术研究院收集整理的大型手写数字数据库。它包含了 6 万张训练图片和 1 万张测试图片,每张图片都是 28×28 像素的灰度图,各自对应 0 到 9 这十个数字类标。

Fashion-MNIST 数据集^[100]由 10 个类别的 70,000

个时尚产品的 28×28 灰度图像组成,每个类别有 7000 张图像。训练集有 6 万张图像,测试集有 1 万张图像。Fashion-MNIST 与 MNIST 具有相同的图像大小、数据格式以及训练和测试分割的结构。

CIFAR-10^[101]由 5 万张 32×32 彩色图像组成。这些图像标有 10 个不同的类标:飞机、汽车、鸟、猫、鹿、狗、青蛙、马、船和卡车。每类有 6000 张图像,5000 张用于训练,1000 张用于测试。与它类似,CIFAR-100^[101]数据集有 100 个类,每个类包含 600 个图像,其中 500 张用于训练,100 张用于测试。

ImageNet 数据集^[33]包含 14197122 个带注释的图像。自 2010 年以来,该数据集用于 ImageNet 大规模视觉识别挑战赛(ILSVRC),这是图像分类和目标检测的基准数据集。Tiny ImageNet^[102]是 ImageNet 数据集的子集,包含 200 个类别的 10 万张彩色图像,尺寸为 64×64 。每类有 500 张训练图像、50 张验证图像和 50 张测试图像。

SVHN 数据集^[103]是一个数字分类基准数据集,包含 60 万张 32×32 RGB 打印的数字(从 0 到 9)图像,由门牌图片中裁剪得出。裁剪后的图像以感兴趣的数字为中心,但保留了附近的数字和其他干扰物。SVHN 有三组:训练集、测试集和一个额外的包含 53 万张图像的集,这些图像难度较低,可用于帮助训练过程。

CelebA 数据集^[104]包含来自 10177 位名人的 202599 张大小为 178×218 的人脸图像,每张都用 40 个二进制标签进行注释,指示面部属性,如头发颜色、性别和年龄。

LFW 数据集^[105]包含从网络收集的 13233 张人脸图像。该数据集由 5749 个身份组成,其中 1680 人有两张或更多张图片。

MS COCO 数据集^[106]是一个大规模的对象检测、分割、关键点检测和字幕数据集,由 32.8 万张图像组成。

表 5 深度学习模型测试中的常用数据集
Table 5 Popular datasets used in testing DNNs

适用场景	数据集	描述	样本数量	类别数	测试目标
图像分类	MNIST ^[32]	手写数字	70000	10	任务性能、安全性
	Fashion-MNIST ^[100]	时尚黑白图	70000	10	任务性能、安全性
	CIFAR-10 ^[101]	10 类通用图片	60000	10	任务性能、安全性
	CIFAR-100 ^[101]	100 类通用图片	60000	100	任务性能、安全性
	ImageNet ^[33]	图像分类竞赛数据集	14197122	20000+	任务性能、安全性
	Tiny ImageNet ^[102]	ImageNet 的子集	100000	200	任务性能、安全性
	SVHN ^[103]	门牌数字	60000	10	任务性能、安全性
	CelebA ^[104]	名人脸部图像	202599	10177	任务性能、安全性
	LFW ^[105]	人脸图像	13233	5749	任务性能、安全性
	MS COCO ^[106]	物体识别	328000	80	任务性能、安全性
语音识别	KITTI ^[107]	交通场景	14999	-	安全性
	VCTK ^[108]	英语口语	44000+	110	安全性
	LibriSpeech ^[109]	有声读物	200	-	安全性
	VoxCeleb ^[110]	名人语音	100000	1251	任务性能、安全性
	SNLI ^[111]	自然语言推理 语料库	60000	-	安全性
自然语言 处理	MultiNLI ^[112]	标注了含义的 句子对	433000	-	安全性
	bAbI ^[113]	问题和对应答案	2000	-	安全性
	Adult ^[114]	人口收入	48842	-	公平性
模型公平 决策	German Credit ^[115]	银行信用风险	1000	-	公平性
	Boston 房价 ^[116]	住房价格	506	-	公平性
	MEPS ^[117]	医疗服务数据	-	-	公平性
	COMPAS ^[118]	再次犯罪可能性	18610	-	公平性
	Bank Marketing ^[119]	银行客户信息	45211	-	公平性
	IMDb ^[120]	电影评论	50000	-	公平性

KITTI 数据集^[107]是用于移动机器人和自动驾驶的最受欢迎的数据集之一。它包括使用各种传感器模式记录的 6 个交通场景, 包括高分辨率 RGB、灰度立体相机和 3D 激光扫描仪。

7.1.2 语音识别

VCTK 语料库^[108]包括 110 位具有各种口音的英语使用者的语音数据。每个发言者读出大约 400 个句子, 这些句子是从报纸、彩虹段落和用于语音重音档案的引出段落中选出的。每位说话者阅读一组不同的报纸句子, 其中每组句子都是使用贪婪算法选择的, 该算法旨在最大化上下文和语音覆盖范围。

LibriSpeech 语料库^[109]包含大约 1000h 的有声读物。训练数据分为 100h、360h 和 500h 集的 3 个分区, 而开发和测试数据分别分为“干净”和“其他”类别, 具体取决于自动语音识别系统的表现是否具有挑战性。每个开发和测试集的音频长度都在 5h 左右。该语料库还提供了从 Project Gutenberg 书籍中摘录的 n-gram 语言模型和相应文本。

VoxCeleb^[110]是一个大型语音识别数据集。它包

含来自 YouTube 视频的 1251 位名人约 10 万段语音。这些名人有不同的口音、职业和年龄。数据基本上是性别平衡的(男性占 55%), 训练集和测试集之间没有重叠。数据集中包括子集 Voxceleb1 和 Voxceleb2。其中 VoxCeleb1 包含超过 10 万个针对 1251 个名人的话语, 这些话语是从上传到 YouTube 的视频短片中提取的。其中发音人数 1251 人、视频数量 21245 条、音频数量 145265 条。Voxceleb2 音频总内容时长达 2000h 以上, 其中发音人数包括训练集 5994 人和测试集 118 人; 音频数量包括训练集 1092009 条和测试集 36237 条。

7.1.3 自然语言处理

SNLI 数据集^[111]由 57 万个手动标记为蕴含、矛盾和中性的句子对组成。注释者判断句子之间的关系, 因为它们描述了相同的事件。每一对都被标记为“entailment”、“neutral”、“contradiction”或“-”, 其中“-”表示无法达成一致。

MultiNLI 数据集^[112]有 43.3 万个句子对。它的大小和收集方式与 SNLI 非常相似。MultiNLI 提供十种

不同类型(面对面、电话、9/11、旅行、信件、牛津大学出版社、批评、逐字、政府和小说)的书面和口语英语数据。

bAbI 数据集^[113]是由 20 个不同任务组成的文本 QA 基准。每项任务旨在测试不同的推理技能, 例如演绎、归纳和共指解析, 其中一些任务需要关系推理。每个样本由一个问题、一个答案和一组事实组成。该数据集有两个不同大小的版本, bAbI-1k 和 bAbI-10k, 后者包含 1 万个训练样本。

7.1.4 模型公平决策

Adult 数据集^[114]含有 48842 个实例, 包括职业、种族、性别、国籍在内的 14 个特征, 可以用于预测一个人的年收入是否超过 5 万美元。

German Credit 数据集^[115]是根据个人的银行贷款信息和申请客户贷款逾期发生情况来预测贷款违约倾向的数据集, 数据集包含 24 个维度的 1000 条数据, 具体包括年龄、工作类型、婚姻状况、性别等 50 个特征。

Boston 房价数据集^[116]包含美国人口普查局收集的美国马萨诸塞州波士顿住房价格的有关信息, 只含有 506 条数据。每条数据包含房屋以及房屋周围的详细信息。其中包含城镇犯罪率、一氧化氮浓度、住宅平均房间数, 到中心区域的加权距离以及自住房平均房价等 14 个特征。

MEPS 数据集^[117]是唯一衡量美国人如何使用和支付医疗保健、健康保险和自付费用的全国性数据来源。其内容包括健康状况、医疗服务使用、收费、保险范围和护理满意度。

COMPAS 数据集^[118]收集了超过 7000 名在佛罗里达州被捕罪犯及其被告再犯的风险。它是美国各地使用的几种风险评估算法之一, 可用于预测暴力犯罪的热点地区, 确定囚犯可能需要的监管类型。目前美国已经有 5 个州靠 COMPAS 来进行刑事司法判决, 其他辖区也已经有其他的风险评估程序就位。

Bank Marketing 数据集^[119]描述了葡萄牙银行营销活动结果, 主要基于直接电话, 向银行客户提供定期存款。该数据集包含 11162 条数据, 50 个特征, 包括年龄、职业、婚姻状况、账户余额等。

IMDb 电影评论数据集^[120]是一个二元情感分析数据集, 由来自互联网电影数据库(IMDb)的 50000 条评论组成, 标记为正面或负面。每部电影包含的评

论不超过 30 条。数据集包含额外的未标记数据。

7.2 模型库

预训练模型库的发展解决了训练成本问题, 拓宽了深度学习模型的应用领域。本节整理了主流的在线模型库, 以便未来的应用和安全性测试。

7.2.1 Caffe Model Zoo

Caffe 是一个考虑了表达、运行速度和模块化的深度学习框架。在 Caffe Model Zoo^①中, 集成了由许多研究人员和工程师使用各种架构和数据为不同的任务制作的 Caffe 模型, 这些预训练模型可以应用于多种任务和研究中, 从简单回归到大规模视觉分类, 再到语音和机器人应用。

7.2.2 ONNX Model Zoo

开放神经网络交换(Open Neural Network Exchange, ONNX)是一种用于表示机器学习模型的开放标准格式。ONNX 定义了一组通用运算符、机器学习和深度学习模型的构建块, 以及一种通用文件格式, 使 AI 开发人员能够使用具有各种框架、工具、运行时和编译器的模型。ONNX Model Zoo^②是由社区成员贡献的 ONNX 格式的预训练的、最先进的集成模型库。模型任务涵盖了图像分类、目标检测、机器翻译等十种多领域任务。

7.2.3 BigML model market

BigML^③是一个可消耗、可编程且可扩展的机器学习平台, 可轻松解决分类、回归、时间序列预报、聚类分析、异常检测、关联发现和主题建模任务, 并将它们自动化。BigML 促进了跨行业的无限预测应用, 包括航空航天、汽车、能源、娱乐、金融服务、食品、医疗保健、物联网、制药、运输、电信等等。

7.2.4 Amazon SageMaker

Amazon SageMaker^④是由亚马逊提供的机器学习服务平台, 通过整合专门为机器学习构建的广泛功能集, 帮助数据科学家和开发人员快速准备、构建、训练和部署高质量的机器学习模型。SageMaker 消除了机器学习过程中每个步骤的繁重工作, 让开发高质量模型变得更加轻松。SageMaker 在单个工具集中提供了用于机器学习的所有组件, 因此模型将可以通过更少的工作量和更低的成本更快地投入生产。

7.3 常用工具包

得益于众多深度学习框架和集成模型库的快速发展, 深度学习模型被广泛应用于安全攸关的领域,

① <https://github.com/BVLC/caffe/wiki/Model-Zoo>

② <https://github.com/onnx/models>

③ <https://bigml.com/>

④ <https://aws.amazon.com/cn/sagemaker/>

测试工具也逐步被开发和完善。本节整理了现有的常用测试工具包。

7.3.1 Themis

Galhotra 等人^[83]提出了 Themis^①, 一个开源的、用于检测因果偏见的公平性测试工具。它可以通过生成有效的测试套件来测量歧视是否存在。在给定描述有效系统输入的模式时, Themis 会自动生成判别测试。其应用场景包括金融贷款、医疗诊断和治疗、促销行为、刑事司法系统等。

7.3.2 mltest

测试工具 mltest^②, 是一个为基于 Tensorflow 的机器学习系统编写单元测试的测试框架。它可以通过极少的设置, 实现包括变量变化、变量恒定、对数范围检查、输入依赖、NaN 和 Inf 张量检查等多种常见的机器学习问题进行综合测试。遗憾的是, Tensorflow2.0 的发布, 破坏了该测试工具的大部分功能。

7.3.3 torchtest

torchtest^③受 mltest 启发, 与 mltest 功能类似, 是为基于 pytorch 的机器学习系统编写单元测试的测试框架。

总体而言, 与传统软件测试不同, 深度学习测试中现有的工具支持发展仍处于初期且尚不成熟。系统且全面的测试工具开发, 存在巨大的发展前景。

8 未来研究方向

目前深度模型的可靠性测试领域的研究工作还有很大的发展空间, 如何更高效地对深度模型进行系统性的测试也成为安全领域相关研究人员的关注点之一。本节根据目前国内外的研究现状, 基于上述对测试任务的介绍, 对深度模型测试的未来发展方向进行探讨。如何构建模型缺陷的多尺度、自适应、可拓展的安全测试框架, 实现在多粒度、多层级上监视、评估极端情况的威胁程度与范围, 是安全可靠人工智能研究中的一个关键问题。

8.1 多领域和多任务

目前不同深度学习类别和任务的测试存在明显的不平衡。在未来的研究工作中, 测试研究应不仅集中于监督学习的分类问题上, 对于日益流行的强化学习和迁移学习的测试方法, 既有挑战, 也有研究机遇。

此外, 当前测试主要围绕卷积神经网络的图像

分类任务展开。在未来的工作中, 研究重心应转移到已有研究甚少的语音识别、自然语言处理等运用循环神经网络等时序分类任务中。设计面向多种场景领域和任务的测试, 也将是未来非常重要的研究方向。

8.2 测试模型组件

目前的研究主要集中于进行数据测试, 但仅从数据安全性考量深度学习模型整体安全性是片面的。研究者应考虑到, 模型安全性的复合效应(如, 高分类精度、强鲁棒性等)是由不同行为组件组合而成, 对于组件的数据、训练程序甚至深度学习框架的测试也是保障测试全面性的解决途径。此外, 在测试过程中, 回归测试、错误成因分析和错误种类划分也应有研究价值。

8.3 测试可解释性

对于测试的可解释性, 现有的方法主要依赖于手动评估。未来的研究中, 应尽可能降低人工成本, 实现自动判断深度学习模型的逻辑或预测结果是否与人类可理解的语义相同。我们认为研究可解释性的自动评估或可解释性违规的检测是十分有必要的, 这在一定程度上保证了测试是否可信。同时, 测试属性的可视化在测试中可以帮助开发人员理解错误, 并帮助错误定位和进一步修复。

8.4 测试代价

在深度模型测试中, 测试代价通常很高, 严重阻碍了测试实际部署模型的可能性。在未来的研究中, 在庞大的测试输入搜索空间以及重复预测过程中降低时间与空间成本十分值得研究。降低成本的一个可能的研究方向是将深度学习模型表示精简表示为某种中间状态, 使其更易于测试。此外, 应用传统的成本降低技术, 如测试优先级或测试样本数目最小化, 以减少测试用量, 同时保持测试的正确性也将是未来缩短测试成本的解决途径。

9 总结

近年来, 从图像识别到自然语言处理, 从社交网络到语音识别深度学习模型在多种领域中得到广泛应用。目前面向深度模型的安全性测试已在自动驾驶和自然语言处理方面等多个实际场景中得到了应用, 然而, 测试技术仍处于初级阶段, 许多关键的科学问题亟待解决。此外, 对于测试方法尚缺少涵盖

① <https://github.com/LASER-UMASS/Themis>

② <https://github.com/Thenerdstation/mltest>

③ <https://github.com/suriyadeepan/torchtest>

多角度、全面完整的中文综述工作。本文概括了面向深度模型的可靠性测试方法, 根据模型任务性能、安全性、公平性和隐私性这四个不同的测试目标, 对现有方法进行了全面的归纳总结, 同时介绍了应用场景和常用数据集、模型库和工具包, 并分析了深度模型可靠性测试未来研究方向, 为设计系统、高效、可信的深度模型测试提供参考。

参考文献

- [1] Bojarski M, Del Testa D, Dworakowski D, et al. End to End Learning for Self-Driving Cars[EB/OL]. 2016: arXiv: 1604.07316. <https://arxiv.org/abs/1604.07316.pdf>.
- [2] Costa-jussà M R. From Feature to Paradigm: Deep Learning in Machine Translation[J]. *Journal of Artificial Intelligence Research*, 2018, 61: 947-974.
- [3] Bhattacharya S, Reddy Maddikunta P K, Pham Q V, et al. Deep Learning and Medical Image Processing for Coronavirus (COVID-19) Pandemic: A Survey[J]. *Sustainable Cities and Society*, 2021, 65: 102589.
- [4] Lambert F. Understanding the fatal tesla accident on autopilot and the nhtsa probe[J]. *Electrek*, July, 2016, 1.
- [5] Wang Z, Yan M, Liu S, et al. Survey on Testing of Deep Neural Networks[J]. *Journal of Software*, 2020, 31(5): 1255-1275.
(王赞, 闫明, 刘爽, 等. 深度神经网络测试研究综述[J]. *软件学报*, 2020, 31(5): 1255-1275.)
- [6] Huang X W, Kroening D, Ruan W J, et al. A Survey of Safety and Trustworthiness of Deep Neural Networks: Verification, Testing, Adversarial Attack and Defence, and Interpretability[J]. *Computer Science Review*, 2020, 37: 100270.
- [7] Li D, Dong C Q, Si P C, et al. Survey of Research on Neural Network Verification and Testing Technology[J]. *Computer Engineering and Applications*, 2021, 57(22): 53-67.
(李舵, 董超群, 司品超, 等. 神经网络验证和测试技术研究综述[J]. *计算机工程与应用*, 2021, 57(22): 53-67.)
- [8] Goodfellow I J, Shlens J, Szegedy C. Explaining and Harnessing Adversarial Examples[EB/OL]. 2014: arXiv: 1412.6572. <https://arxiv.org/abs/1412.6572.pdf>.
- [9] Szegedy C, Zaremba W, Sutskever I, et al. Intriguing Properties of Neural Networks[EB/OL]. 2013: arXiv: 1312.6199. <https://arxiv.org/abs/1312.6199.pdf>.
- [10] Carlini N, Wagner D. Towards Evaluating the Robustness of Neural Networks[C]. *2017 IEEE Symposium on Security and Privacy*, 2017: 39-57.
- [11] Dong Y P, Liao F Z, Pang T Y, et al. Boosting Adversarial Attacks with Momentum[C]. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018: 9185-9193.
- [12] Gu T Y, Dolan-Gavitt B, Garg S. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain[EB/OL]. 2017: arXiv: 1708.06733. <https://arxiv.org/abs/1708.06733.pdf>.
- [13] Shafahi A, Huang W R, Najibi M, et al. Poison frogs! targeted clean-label poisoning attacks on neural networks[EB/OL]. 2018: ArXiv Preprint ArXiv:1804.00792.
- [14] Saha A, Subramanya A, Pirsiavash H. Hidden Trigger Backdoor Attacks[J]. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, 34(7): 11957-11965.
- [15] Salem A, Wen R, Backes M, et al. Dynamic Backdoor Attacks Against Machine Learning Models[EB/OL]. 2020: arXiv: 2003.03675. <https://arxiv.org/abs/2003.03675.pdf>.
- [16] Pei K X, Cao Y Z, Yang J F, et al. DeepXplore: Automated White-box Testing of Deep Learning Systems[EB/OL]. 2017: arXiv: 1705.06640. <https://arxiv.org/abs/1705.06640.pdf>.
- [17] Klees G, Ruef A, Cooper B, et al. Evaluating Fuzz Testing[C]. *The 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018: 2123-2138.
- [18] Ma L, Zhang F Y, Sun J Y, et al. DeepMutation: Mutation Testing of Deep Learning Systems[C]. *2018 IEEE 29th International Symposium on Software Reliability Engineering*, 2018: 100-111.
- [19] Ma L, Juefei-Xu F, Xue M H, et al. DeepCT: Tomographic Combinatorial Testing for Deep Learning Systems[C]. *2019 IEEE 26th International Conference on Software Analysis, Evolution and Re-engineering*, 2019: 614-618.
- [20] Baldoni R, Coppa E, D'elia D C, et al. A Survey of Symbolic Execution Techniques[J]. *ACM Computing Surveys*, 51(3)Article No. 50.
- [21] Sun Y C, Wu M, Ruan W J, et al. Concolic Testing for Deep Neural Networks[C]. *The 33rd ACM/IEEE International Conference on Automated Software Engineering*, 2018: 109-119.
- [22] Japkowicz N. Why question machine learning evaluation methods[C]. *AAAI workshop on evaluation methods for machine learning*. 2006: 6-11.
- [23] Chen W J, Gallas B D, Yousef W A. Classifier Variability: Accounting for Training and Testing[J]. *Pattern Recognition*, 2012, 45(7): 2661-2671.
- [24] Chen W J, Samuelson F W, Gallas B D, et al. On the Assessment of the Added Value of New Predictive Biomarkers[J]. *BMC Medical Research Methodology*, 2013, 13(1): 1-9.
- [25] Qin Y, Wang H Y, Xu C, et al. SynEva: Evaluating ML Programs by Mirror Program Synthesis[C]. *2018 IEEE International Conference on Software Quality, Reliability and Security*, 2018: 171-182.
- [26] Zhang J M, Barr E T, Guedj B, et al. Perturbed Model Validation: A New Framework to Validate Model Relevance[EB/OL]. 2019: arXiv: 1905.10201. <https://arxiv.org/abs/1905.10201.pdf>.
- [27] Werpachowski R, György A, Szepesvári C. Detecting Overfitting via Adversarial Examples[EB/OL]. 2019: arXiv: 1903.02380. <https://arxiv.org/abs/1903.02380.pdf>.
- [28] Chatterjee S, Mishchenko A. Circuit-Based Intrinsic Methods to Detect Overfitting[EB/OL]. 2019: arXiv: 1907.01991. <https://arxiv.org/abs/1907.01991.pdf>.
- [29] Gossman A, Pezeshek A, Sahiner B. Test Data Reuse for Evaluation of Adaptive Machine Learning Algorithms: Over-Fitting to a Fixed 'Test' Dataset and a Potential Solution[C]. *SPIE Medical Imaging. Proc SPIE 10577, Medical Imaging 2018: Image Perception, Observer Performance, and Technology Assessment*, 2018, 10577: 121-132.
- [30] Ma S Q, Liu Y Q, Lee W C, et al. MODE: Automated Neural

- Network Model Debugging via State Differential Analysis and Input Selection[C]. *The 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2018: 175-186.
- [31] Guo J M, Jiang Y, Zhao Y, et al. DLFuzz: Differential Fuzzing Testing of Deep Learning Systems[C]. *The 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2018: 739-743.
- [32] LeCun Y, Bottou L, Bengio Y, et al. Gradient-Based Learning Applied to Document Recognition[J]. *Proceedings of the IEEE*, 1998, 86(11): 2278-2324.
- [33] Deng J, Dong W, Socher R, et al. ImageNet: A Large-Scale Hierarchical Image Database[C]. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009: 248-255.
- [34] Odena A, Goodfellow I. TensorFuzz: Debugging Neural Networks with Coverage-Guided Fuzzing[EB/OL]. 2018: arXiv: 1807.10875. <https://arxiv.org/abs/1807.10875.pdf>.
- [35] Tian Y C, Pei K X, Jana S, et al. DeepTest: Automated Testing of Deep-Neural-Network-Driven Autonomous Cars[C]. *2018 IEEE/ACM 40th International Conference on Software Engineering*, 2018: 303-314.
- [36] Ma L, Juefei-Xu F, Zhang F Y, et al. DeepGauge: Multi-Granularity Testing Criteria for Deep Learning Systems[C]. *The 33rd ACM/IEEE International Conference on Automated Software Engineering*, 2018: 120-131.
- [37] Xie X F, Ma L, Juefei-Xu F, et al. DeepHunter: A Coverage-Guided Fuzz Testing Framework for Deep Neural Networks[C]. *The 28th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2019: 146-157.
- [38] Lee S, Cha S, Lee D, et al. Effective White-Box Testing of Deep Neural Networks with Adaptive Neuron-Selection Strategy[C]. *The 29th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2020: 165-176.
- [39] Sun Y C, Huang X W, Kroening D, et al. Testing Deep Neural Networks[EB/OL]. 2018: arXiv: 1803.04792. <https://arxiv.org/abs/1803.04792.pdf>.
- [40] Du X N, Xie X F, Li Y, et al. DeepCruiser: Automated Guided Testing for Stateful Deep Learning Systems[EB/OL]. 2018: arXiv: 1812.05339. <https://arxiv.org/abs/1812.05339.pdf>.
- [41] Du X N, Xie X F, Li Y, et al. DeepStellar: Model-Based Quantitative Analysis of Stateful Deep Learning Systems[C]. *The 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2019: 477-487.
- [42] Huang W, Sun Y C, Huang X W, et al. TestRNN: Coverage-Guided Testing on Recurrent Neural Networks[EB/OL]. 2019: arXiv: 1906.08557. <https://arxiv.org/abs/1906.08557.pdf>.
- [43] Li Z N, Ma X X, Xu C, et al. Structural Coverage Criteria for Neural Networks could be Misleading[C]. *2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results*, 2019: 89-92.
- [44] Harel-Canada F, Wang L X, Ali Gulzar M, et al. Is Neuron Coverage a Meaningful Measure for Testing Deep Neural Networks? [C]. *The 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2020: 851-862.
- [45] Dong Y Z, Zhang P X, Wang J Y, et al. There is Limited Correlation between Coverage and Robustness for Deep Neural Networks[EB/OL]. 2019: arXiv: 1911.05904. <https://arxiv.org/abs/1911.05904.pdf>.
- [46] Jia Y, Harman M. An Analysis and Survey of the Development of Mutation Testing[J]. *IEEE Transactions on Software Engineering*, 2011, 37(5): 649-678.
- [47] Hu Q, Ma L, Xie X F, et al. DeepMutation: A Mutation Testing Framework for Deep Learning Systems[C]. *2019 34th IEEE/ACM International Conference on Automated Software Engineering*, 2020: 1158-1161.
- [48] Humbatova N, Jahangirova G, Tonella P. DeepCrime: Mutation Testing of Deep Learning Systems Based on Real Faults[C]. *The 30th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2021: 67-78.
- [49] Shen W J, Wan J, Chen Z Y. MuNN: Mutation Analysis of Neural Networks[C]. *2018 IEEE International Conference on Software Quality, Reliability and Security Companion*, 2018: 108-115.
- [50] Riccio V, Humbatova N, Jahangirova G, et al. DeepMetis: Augmenting a Deep Learning Test Set to Increase Its Mutation Score[EB/OL]. 2021: arXiv: 2109.07514. <https://arxiv.org/abs/2109.07514.pdf>.
- [51] Zhang H, Chan W K. Apricot: A Weight-Adaptation Approach to Fixing Deep Learning Models[C]. *2019 34th IEEE/ACM International Conference on Automated Software Engineering*, 2020: 376-387.
- [52] Zhang H, Chan W K. Plum: Exploration and Prioritization of Model Repair Strategies for Fixing Deep Learning Models[C]. *2021 8th International Conference on Dependable Systems and Their Applications*, 2021: 140-151.
- [53] Borkar T S, Karam L J. DeepCorrect: Correcting DNN Models Against Image Distortions[J]. *IEEE Transactions on Image Processing*, 2019, 28(12): 6022-6034.
- [54] Eniser H F, Gerasimou S, Sen A. DeepFault: Fault Localization for Deep Neural Networks[EB/OL]. 2019: arXiv: 1902.05974. <https://arxiv.org/abs/1902.05974.pdf>.
- [55] Wang J Y, Chen J L, Sun Y C, et al. RobOT: Robustness-Oriented Testing for Deep Learning Systems[C]. *2021 IEEE/ACM 43rd International Conference on Software Engineering*, 2021: 300-311.
- [56] Gao X, Saha R K, Prasad M R, et al. Fuzz Testing Based Data Augmentation to Improve Robustness of Deep Neural Networks[C]. *The ACM/IEEE 42nd International Conference on Software Engineering*, 2020: 1147-1158.
- [57] Liu Z X, Feng Y, Chen Z Y. DialTest: Automated Testing for Recurrent-Neural-Network-Driven Dialogue Systems[C]. *The 30th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2021: 115-126.
- [58] Yu B, Qi H, Guo Q, et al. DeepRepair: Style-Guided Repairing for DNNs in the Real-World Operational Environment[EB/OL]. 2020: arXiv: 2011.09884. <https://arxiv.org/abs/2011.09884.pdf>.
- [59] Hou Y H, Liu J W, Wang D W, et al. TauMed: Test Augmentation

- of Deep Learning in Medical Diagnosis[C]. *The 30th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2021: 674-677.
- [60] Wang B L, Yao Y S, Shan S, et al. Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks[C]. *2019 IEEE Symposium on Security and Privacy*, 2019: 707-723.
- [61] Guo W B, Wang L, Xing X Y, et al. TABOR: A Highly Accurate Approach to Inspecting and Restoring Trojan Backdoors in AI Systems[EB/OL]. 2019: arXiv: 1908.01763. <https://arxiv.org/abs/1908.01763.pdf>.
- [62] Harikumar H, Le V, Rana S T, et al. Scalable Backdoor Detection in Neural Networks[EB/OL]. 2020: arXiv: 2006.05646. <https://arxiv.org/abs/2006.05646.pdf>.
- [63] Chen H, Fu C, Zhao J, et al. DeepInspect: A Black-box Trojan Detection and Mitigation Framework for Deep Neural Networks[C]. *IJCAI*. 2019: 4658-4664.
- [64] Tang D, Wang X F, Tang H X, et al. Demon in the Variant: Statistical Analysis of DNNs for Robust Backdoor Contamination Detection[EB/OL]. 2019: arXiv: 1908.00686. <https://arxiv.org/abs/1908.00686.pdf>.
- [65] Xu X J, Wang Q, Li H C, et al. Detecting AI Trojans Using Meta Neural Analysis[C]. *2021 IEEE Symposium on Security and Privacy*, 2021: 103-120.
- [66] Liu Y Q, Lee W C, Tao G H, et al. ABS: Scanning Neural Networks for Back-Doors by Artificial Brain Stimulation[C]. *The 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019: 1265-1282.
- [67] Liu Y Q, Shen G Y, Tao G H, et al. EX-RAY: Distinguishing Injected Backdoor from Natural Features in Neural Networks by Examining Differential Feature Symmetry[EB/OL]. 2021: arXiv: 2103.08820. <https://arxiv.org/abs/2103.08820.pdf>.
- [68] Feng Y, Shi Q K, Gao X Y, et al. DeepGini: Prioritizing Massive Tests to Enhance the Robustness of Deep Neural Networks[C]. *The 29th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2020: 177-188.
- [69] Byun T, Sharma V, Vijayakumar A, et al. Input Prioritization for Testing Neural Networks[C]. *2019 IEEE International Conference on Artificial Intelligence Testing*, 2019: 63-70.
- [70] Zhang L, Sun X C, Li Y, et al. A Noise-Sensitivity-Analysis-Based Test Prioritization Technique for Deep Neural Networks[EB/OL]. 2019: arXiv: 1901.00054. <https://arxiv.org/abs/1901.00054.pdf>.
- [71] Zhang K, Zhang Y T, Zhang L W, et al. Neuron Activation Frequency Based Test Case Prioritization[C]. *2020 International Symposium on Theoretical Aspects of Software Engineering*, 2021: 81-88.
- [72] Ma W, Papadakis M, Tsakmalis A, et al. Test Selection for Deep Learning Systems[J]. *ACM Transactions on Software Engineering and Methodology*, 30(2)Article No. 13.
- [73] Wang Z, You H M, Chen J J, et al. Prioritizing Test Inputs for Deep Neural Networks via Mutation Analysis[C]. *2021 IEEE/ACM 43rd International Conference on Software Engineering*, 2021: 397-409.
- [74] Kim J, Feldt R, Yoo S. Guiding Deep Learning System Testing Using Surprise Adequacy[C]. *2019 IEEE/ACM 41st International Conference on Software Engineering*, 2019: 1039-1049.
- [75] Shen W J, Li Y H, Chen L, et al. Multiple-Boundary Clustering and Prioritization to Promote Neural Network Retraining[C]. *The 35th IEEE/ACM International Conference on Automated Software Engineering*, 2020: 410-422.
- [76] Meng L H, Li Y H, Chen L, et al. Measuring Discrimination to Boost Comparative Testing for Multiple Deep Learning Models[C]. *2021 IEEE/ACM 43rd International Conference on Software Engineering*, 2021: 385-396.
- [77] Guerriero A, Pietrantuono R, Russo S. Operation is the Hardest Teacher: Estimating DNN Accuracy Looking for Mispredictions[C]. *2021 IEEE/ACM 43rd International Conference on Software Engineering*, 2021: 348-358.
- [78] Udeshi S, Arora P, Chattopadhyay S. Automated Directed Fairness Testing[C]. *The 33rd ACM/IEEE International Conference on Automated Software Engineering*, 2018: 98-108.
- [79] Agarwal A, Lohia P, Nagar S, et al. Automated Test Generation to Detect Individual Discrimination in AI Models[EB/OL]. 2018: arXiv: 1809.03260. <https://arxiv.org/abs/1809.03260.pdf>.
- [80] Zhang P X, Wang J Y, Sun J, et al. White-Box Fairness Testing through Adversarial Sampling[C]. *2020 IEEE/ACM 42nd International Conference on Software Engineering*, 2020: 949-960.
- [81] Zhang L F, Zhang Y L, Zhang M. Efficient White-Box Fairness Testing through Gradient Search[C]. *The 30th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2021: 103-114.
- [82] Zheng H B, Chen Z Q, Du T Y, et al. NeuronFair: Interpretable White-Box Fairness Testing through Biased Neuron Identification[EB/OL]. 2021: arXiv: 2112.13214. <https://arxiv.org/abs/2112.13214.pdf>.
- [83] Galhotra S, Brun Y, Meliou A. Fairness Testing: Testing Software for Discrimination[C]. *The 2017 11th Joint Meeting on Foundations of Software Engineering*, 2017: 498-510.
- [84] Sun B, Sun J, Pham H L, et al. Causality-Based Neural Network Repair[EB/OL]. 2022: arXiv: 2204.09274. <https://arxiv.org/abs/2204.09274.pdf>.
- [85] Bichsel B, Gehr T, Drachsler-Cohen D, et al. DP-Finder: Finding Differential Privacy Violations by Sampling and Optimization[C]. *The 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018: 508-524.
- [86] Ding Z Y, Wang Y X, Wang G H, et al. Detecting Violations of Differential Privacy[C]. *The 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018: 475-489.
- [87] Zhang H C, Roth E, Haeberlen A, et al. Testing Differential Privacy with Dual Interpreters[J]. *Proceedings of the ACM on Programming Languages*, 2020, 4(OOPSLA)Article No. 165.
- [88] Wang Y X, Ding Z Y, Kifer D, et al. CheckDP: An Automated and Integrated Approach for Proving Differential Privacy or Finding Precise Counterexamples[C]. *The 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020: 919-938.
- [89] Bichsel B, Steffen S, Bogunovic I, et al. DP-Sniper: Black-Box Discovery of Differential Privacy Violations Using Classifiers[C]. *2021 IEEE Symposium on Security and Privacy*, 2021: 391-409.
- [90] Farina G P, Chong S, Gaboardi M. Coupled Relational Symbolic Execution for Differential Privacy[EB/OL]. 2020: arXiv:

- 2007.12987. <https://arxiv.org/abs/2007.12987.pdf>.
- [91] Zhang M S, Zhang Y Q, Zhang L M, et al. DeepRoad: GAN-Based Metamorphic Testing and Input Validation Framework for Autonomous Driving Systems[C]. *The 33rd ACM/IEEE International Conference on Automated Software Engineering*, 2018: 132-142.
- [92] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative Adversarial Networks[J]. *Communications of the ACM*, 2020, 63(11): 139-144.
- [93] Zhou H S, Li W, Kong Z L, et al. DeepBillboard: Systematic Physical-World Testing of Autonomous Driving Systems[C]. *The ACM/IEEE 42nd International Conference on Software Engineering*, 2020: 347-358.
- [94] Riccio V, Tonella P. Model-Based Exploration of the Frontier of Behaviours for Deep Learning System Testing[C]. *The 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2020: 876-888.
- [95] Birchler C, Khatiri S, Derakhshanfar P, et al. Automated Test Cases Prioritization for Self-driving Cars in Virtual Environments[EB/OL]. 2021: ArXiv Preprint ArXiv:2107.09614.
- [96] Asyofi M H, Thung F, Lo D, et al. CrossASR: Efficient Differential Testing of Automatic Speech Recognition via Text-to-Speech[C]. *2020 IEEE International Conference on Software Maintenance and Evolution*, 2020: 640-650.
- [97] Asyofi M H, Yang Z, Lo D. CrossASR++: A Modular Differential Testing Framework for Automatic Speech Recognition[EB/OL]. 2021: ArXiv Preprint ArXiv:2105.14881.
- [98] Sun L Q, Zhou Z Q. Metamorphic Testing for Machine Translations: MT4MT[C]. *2018 25th Australasian Software Engineering Conference*, 2018: 96-100.
- [99] Sun Z Y, Zhang J M, Harman M, et al. Automatic Testing and Improvement of Machine Translation[C]. *The ACM/IEEE 42nd International Conference on Software Engineering*, 2020: 974-985.
- [100] Xiao H, Rasul K, Vollgraf R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms[EB/OL]. 2017: ArXiv Preprint ArXiv:1708.07747.
- [101] Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images. Technical report, 2009.
- [102] Le Y, Yang X. Tiny imagenet visual recognition challenge[J]. *CS231N*, 2015, 7(7): 3.
- [103] Netzer Y, Wang T, Coates A, et al. Reading digits in natural images with unsupervised feature learning[C]. *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011: 1-9.
- [104] Liu Z W, Luo P, Wang X G, et al. Deep Learning Face Attributes in the Wild[C]. *2015 IEEE International Conference on Computer Vision*, 2016: 3730-3738.
- [105] Huang G B, Mattar M, Berg T, et al. Labeled faces in the wild: A database for studying face recognition in unconstrained environments[C]. *Workshop on faces in 'Real-Life' Images: detection, alignment, and recognition*. 2008: 1-15.
- [106] Lin T Y, Maire M, Belongie S, et al. Microsoft COCO: Common Objects in Context[EB/OL]. 2014: arXiv: 1405.0312. <https://arxiv.org/abs/1405.0312.pdf>.
- [107] Geiger A, Lenz P, Urtasun R. Are we Ready for Autonomous Driving? the KITTI Vision Benchmark Suite[C]. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012: 3354-3361.
- [108] Veaux C, Yamagishi J, MacDonald K. CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit. Technical report. University of Edinburgh. The Centre for Speech Technology Research (CSTR), 2017.
- [109] Panayotov V, Chen G G, Povey D, et al. Librispeech: An ASR Corpus Based on Public Domain Audio Books[C]. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015: 5206-5210.
- [110] Nagrani A, Chung J S, Zisserman A. VoxCeleb: A Large-Scale Speaker Identification Dataset[EB/OL]. 2017: arXiv: 1706.08612. <https://arxiv.org/abs/1706.08612.pdf>.
- [111] Bowman S R, Angeli G, Potts C, et al. A Large Annotated Corpus for Learning Natural Language Inference[EB/OL]. 2015: arXiv: 1508.05326. <https://arxiv.org/abs/1508.05326.pdf>.
- [112] Williams A, Nangia N, Bowman S R. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference[EB/OL]. 2017: arXiv: 1704.05426. <https://arxiv.org/abs/1704.05426.pdf>.
- [113] Weston J, Bordes A, Chopra S, et al. Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks[EB/OL]. 2015: arXiv: 1502.05698. <https://arxiv.org/abs/1502.05698.pdf>.
- [114] Kohavi R. Scaling up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid[C]. *The Second International Conference on Knowledge Discovery and Data Mining*, 1996: 202-207.
- [115] German Credit Datasets. <http://archive.ics.uci.edu/ml/machine-learning-databases/statlog/german/>. 2000.
- [116] The Boston Housing Dataset. <https://www.cs.toronto.edu/~delve/data/boston/bostonDetail.html>. 1978.
- [117] Cohen S B. Design Strategies and Innovations in the Medical Expenditure Panel Survey[J]. *Medical Care*, 2003, 41: III-5-III-12.
- [118] Zafar M B, Valera I, Rodriguez M G, et al. Fairness beyond Disparate Treatment & Disparate Impact: Learning Classification without Disparate Mistreatment[C]. *The 26th International Conference on World Wide Web*, 2017: 1171-1180.
- [119] Moro S, Cortez P, Rita P. A Data-Driven Approach to Predict the Success of Bank Telemarketing[J]. *Decision Support Systems*, 2014, 62: 22-31.
- [120] Maas A L, Daly R E, Pham P T, et al. Learning Word Vectors for Sentiment Analysis[C]. *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, 2011: 142-150.



陈若曦 于 2020 年在浙江工业大学电气工程及其自动化专业获得学士学位。现在浙江工业大学控制工程专业攻读博士学位。研究领域为人工智能安全。研究兴趣包括: 对抗攻防、深度模型测试。Email: 2112003149@zjut.edu.cn.



金海波 于 2020 年在浙江工业大学通信工程专业获得学士学位。现在浙江工业大学控制科学与工程专业攻读博士学位。研究领域为深度学习、人工智能安全。研究兴趣包括: 对抗攻防、深度模型测试。Email: 2112003035@zjut.edu.cn.



陈晋音 于 2009 年在浙江工业大学控制科学与工程专业获得博士学位。现任浙江工业大学网络空间安全研究院教授。研究领域为人工智能、数据挖掘、智能计算。研究兴趣包括: 可信人工智能技术及其应用中的安全问题。Email: chenjinyin@zjut.edu.cn.



郑海斌 分别于 2017 年和 2022 年在浙江工业大学电气工程及其自动化专业和控制科学与工程专业获得学士和博士学位。现任浙江工业大学网络安全研究院讲师。研究领域为深度学习、人工智能安全。研究兴趣包括: 对抗攻防、深度模型公平性。Email: haibinzheng320@gmail.com.



李晓豪 于 2020 年在常熟理工学院自动化专业获得工学学士学位。现在浙江工业大学控制科学与工程专业攻读硕士研究生学位。研究领域为人工智能安全、网络安全。研究兴趣包括网络攻防、深度学习等。Email: xiaohaoli0124@gmail.com.