arXiv:2303.01295v1 [cs.LG] 2 Mar 2023

深度神经网络运行准确率的迭代评估与改进

Antonio Guerriero、Roberto Pietrantuono、Stefano Russo*DIETI*，那不勒斯腓特烈二世大学克劳迪奥街21号，80125 - 那不勒斯，意大利 {antonio.guerriero, roberto.pietrantuono,stefano.russo}@unina.it

摘要—**深度神经网络（深度神经网络）如今广泛应用于许多领域，这得益于它们在人类特定任务上近似甚至超越人类的表现。然而，由于运行条件不可预测或未曾考虑，实际应用中会出现意外故障，使得深度神经网络在运行中的表现与发布前估计的表现大不相同。**

在深度神经网络系统生命周期中，准确率评估通常通过两种方式进行：离线方式，即对运行输入进行抽样；或在线方式，即通过伪预言机进行评估。前者被认为成本较高，因为需要对抽样得到的输入进行人工标注。后者是自动的，但准确率较低。

我们认为，新兴的机器学习系统迭代式工业级生命周期模型（如机器学习运维），不仅能够利用运行中观察到的输入，准确估计深度神经网络的准确率，还能通过重构/重新训练来提升准确率。

我们提出 DAIC（深度神经网络评估与改进周期），这是一种结合“低成本”在线伪预言机和“高成本”离线抽样技术的方法，旨在深度神经网络生命周期的各次迭代中估计并提升其运行准确率。初步结果表明，结合这两种方法并将其整合到深度神经网络生命周期中具有显著优势。

索引词—**深度神经网络、准确率评估、准确率提升**

一、引言

如今，机器学习在各种应用领域得到广泛采用。这一趋势得益于机器学习，尤其是深度神经网络，在许多任务中达到人类水平的能力 [1]–[3]。

机器学习系统的可靠性通常以准确率来衡量。对于分类任务，准确率是正确分类的示例数占总示例数的比例。自动评估深度神经网络准确率的难题，仍然对其在关键领域的应用构成威胁。

与评价准确率并进而提升深度神经网络相关的主要活动，通常在其发布到执行环境之前进行。蜕变测试 [4]和变异测试 [5], [6] 是评价深度神经网络鲁棒性并预测这些系统在运行环境中可靠性的最常见策略。

然而，发布前估计的准确率可能与运行期间获得的准确率（运行准确率）存在显著差异。Retch 等人 展示了当提交完全不同的新数据时，分类器的准确率分数可能会大幅下降 [7]。这一问题日益严重，尤其是在运行过程中出现意外现象时，例如分布偏移 或 标签偏移 [8]。

针对深度神经网络的迭代生命周期（例如机器学习运维 [9],[10]）已由 Google 等公司提出。在这些深度神经网络生命周期模型中，开发阶段与运行阶段通过一个循环相连 [11], ，旨在根据运行条件评价并提升深度神经网络的准确率。具体而言，这些模型可以利用运行数据，在新一轮部署之前重构和/或重新训练深度神经网络（实验阶段），并在实际运行中自动评价准确率以及自动重新训练模型（部署阶段）。

通过监控深度神经网络收集到的运行数据，其真实标签通常是未知的。这是软件测试中的一个普遍问题，称为预言机问题 [12]。对于机器学习系统，根据 Murphy等人的说法，“对于任意输入，都不存在可靠的测试预言机来指出正确的输出应该是什么” [13]。这个问题显然也影响深度神经网络准确率估计 [14]。

评估深度神经网络运行准确率有两种方法：i）借助伪预言机 （通常也基于机器学习模型）自动评估运行输入是否分类正确；伪预言机可依据多种知识来源检测错误预测；ii）通过对少量具有代表性的输入进行适当的统计抽样，缩减待标注运行数据集的规模。

伪预言机无需人工干预；然而，由于用于评估待评估深度神经网络输出的知识具有概率性，它们通常会产生大量假阳性[15],。抽样技术可以减少成本高昂且耗时的人工预言机反馈，但无法完全避免；不过，它们能够避免假阳性，并更忠实地估算深度神经网络运行准确率。

本文提出深度神经网络评估与改进周期（DAIC），将通过伪预言机进行的自动评估与抽样评估相结合。其目标是在降低人工干预成本的同时，忠实地估算深度神经网络的运行准确率，并利用新获得的有标签样本采取重构／重新训练措施，以提高深度神经网络准确率。

MNIST 手写数字数据集上的初步实验结果 [16] 表明，DAIC 能够有效地

利用自动伪预言机，通过未标注样本跟踪深度神经网络的准确率，并仅在必要时触发高成本的基于抽样的评估以更新准确率估计。随后，进一步利用收集到的运行样本，在迭代周期中提高深度神经网络的准确率。DAIC 能够稳健应对标签偏移等现象。

本文结构如下。第二节介绍深度神经网络运行准确率评估技术。第三节介绍深度神经网络评估与改进周期；第四节展示初步结果。第五节介绍未来计划；第六节给出结论。

II. 深度神经网络运行准确率评估

A. 通过伪预言机进行评估

自动伪预言机通常通过基于训练集所编码知识的交叉引用 [17]–[19] 来构建。这些知识通过多个彼此不同的实现提取（例如，采用不同的机器学习模型，或采用架构不同的同一机器学习模型），以进行多数投票。这些技术会受到训练集偏差的严重影响。当训练数据不能代表运行环境时，这些预言机的性能会显著下降。

人们还提出了其他技术，以从训练数据中提取知识来构建自动预言机，例如使用专用网络（ConfidNet [20] 和自编码器[21]），或利用被评估系统自身的特征（例如内部层的输出 [22]）。

仅将训练数据集和深度神经网络作为知识来构建伪预言机的技术，对运行环境偏离部署前环境尤为敏感。因此，遇到标签偏移等现象时，这些技术预计表现不佳 [8], [23]。当输入标签的分布相对于训练时发生变化，而其他一切保持不变时，监督式深度神经网络算法就会面临标签偏移：在实践中，这意味着未标注的运行输入与训练样本相似，因此深度神经网络会按照训练时的方式对其进行分类，但它们的实际类别却与训练期间学到的类别不同。

针对图像分类问题，已有研究提出 ICOS 预言机替代方案，用于评估卷积神经网络在投入运行后 [24] 的准确率。ICOS 从不同知识来源中提取不变量，以评估未标注的运行样本。

与 ICOS 类似，我们考虑一种伪预言机——下文称为 DNN-OS——它利用三种不同的知识来源（运行领域、训练数据和深度神经网络）来定义三组不变量（领域、数据、模型），用于自动评估待评估深度神经网络的输出。

一个领域不变量的例子适用于自动驾驶车辆：假设道路限速为 50km/h，则：$fail$:- $speed_limit = 50\ km/h,\ accelerate = true,\ current_speed = 50\ km/h$。此类领域不变量使预言机能够通过检查深度神经网络的输出及其影响来检测故障以及其对整个系统的影响。与最先进技术相比，使用领域不变量能使 DNN-OS 在运行过程中抵御意外现象。

数据和模型不变式可以通过机器学习算法从训练数据集和验证数据集中自动提取。这些不变式分别在输入数据和深度神经网络中寻找模式，例如，发生故障时始终呈现特定值的像素子集（对于数据）或神经元子集（对于模型）。

由于其特性，通过伪预言机进行的评估可以在线执行，也就是在系统运行期间执行。自动化预言机会根据实际输入计算准确率。此估算可用于提示测试人员是否需要采取纠正或改进措施。

在线评估的特点是，伪预言机算法的“知识提取”和参数调优需要一次性支付固定成本。

B. 通过抽样进行评估

近期文献 [25]–[28] 探讨了如何通过抽样降低运行样本人工标注的成本。一些技术用于选取一个能够准确代表总体的小型数据样本[25]–[27]，以便对运行期间达到的准确率进行可靠估算。代表性样本中导致错误预测的示例比例应大致与运行数据集中的比例相同。

然而，仅仅模仿预期输入可能效率低下，尤其是对于准确率很高的深度神经网络，因为要对正确分类的示例进行人工标注，以获得可接受的运行准确率估计，需要付出很大努力。因此，显而易见，更可取的做法是尽可能多地抽取与错误输出相关的示例，同时仍能对运行准确率做出无偏估计。

借助 DeepEST，Guerriero 等人 [28] 旨在既提供可靠的运行准确率估计，又尽可能多地抽取故障样本（例如错误分类），并在估计过程中对不均衡抽样进行校正。

这一评估策略可以离线执行，也就是说，在可获得监测到的运行数据及深度神经网络输出时进行。利用这些数据，可以通过抽样和人工标注来估算准确率。人工标注运行输入的高昂成本，可通过利用已标注示例对受评估的深度神经网络采取改进措施来抵消。

III. 深度神经网络评估与改进周期

一种降低应用成本并最大化收益的方法，是将在线评估与离线评估结合成一个周期，称为深度神经网络评估与改进周期（DAIC）。其理念是，在每个周期通过伪预言机获得一个“低成本”的准确率估计值，并且仅当自动伪预言机估计的运行准确率低于给定阈值时，才触发“高成本”（但更为可靠）的基于离线抽样的估计。

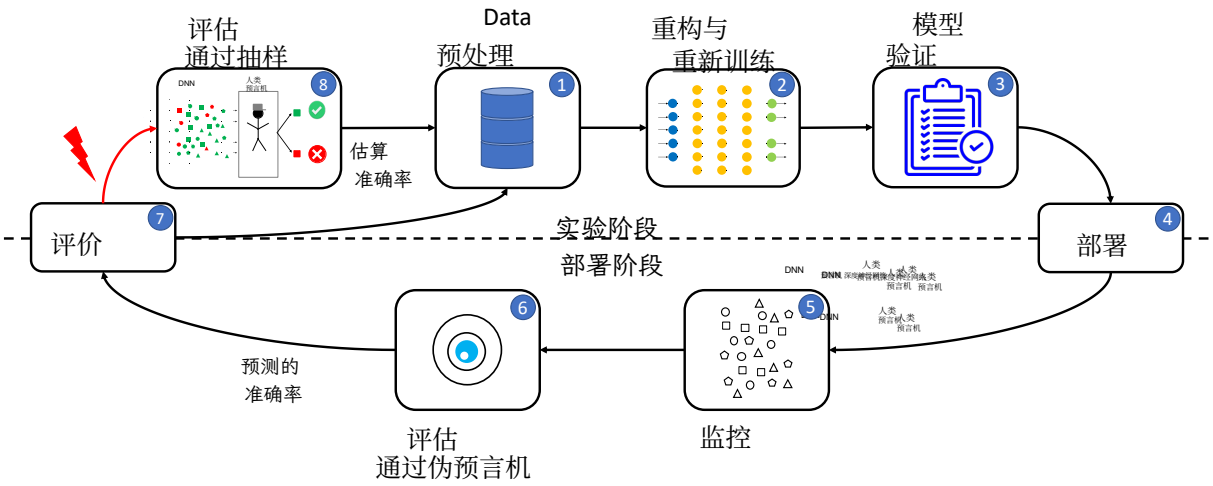

图 1. 提出的类机器学习运维深度神经网络评估与改进周期（DAIC）

与机器学习运维一样，DAIC 包含一个实验阶段和一个部署阶段，具体包括以下阶段（图 1）：

1) **数据预处理**：在一个周期的起始阶段，根据新增的有标签样本（如果有）以及上一轮迭代中计算出的准确率估计值，更新训练数据集和验证数据集。此阶段会更新训练集，使其更好地反映实际观测到的运行条件。2) **重新建模与再训练**：从头开始训练模型（在第一次迭代或进行重构时），或者使用数据预处理阶段输出的训练集重新训练模型。3) **模型验证**：在发布之前，使用由数据预处理阶段生成的验证数据集，计算已训练深度神经网络的准确率（验证准确率）。4) **部署**：将深度神经网络部署到执行环境中并投入运行。5) **监控**：收集输入深度神经网络的（无标签）输入及相应的深度神经网络输出结果；如果可用，还可以收集有关运行条件的其他信息（输入源、用户类型、运行配置等），以构建领域不变量。6) **通过伪预言机进行评估**：使用自动伪预言机将深度神经网络的每个输出分类为通过或失败。利用预言机的预测结果计算深度神经网络运行时准确率的估计值，称为预测准确率。7) **评价**：当预测准确率与验证准确率出现偏差时，将触发离线评估会话（8）；否则跳过基于抽样的评估。8) **通过抽样进行评估**：抽取一组输入并进行（人工）标注，然后计算运行准确率的估计值。

其理念是利用伪预言机持续评价深度神经网络提供的运行准确率，从而降低人工标注、重新训练和重构的成本，仅在需要时执行这些操作。

IV. 实验

A. 准确率评估算法、数据集、开放工件

DAIC 实验采用两个伪预言机和一种抽样准确率评估算法进行。

这两个伪预言机分别是 SelfChecker [22]（一种自动预言机，它利用被测深度神经网络自身的特征来评估预测结果）和 DNN-OS。

基于抽样的评估算法是 DeepEST [28]；它会考虑辅助变量（例如深度神经网络预测结果的置信度），以引导抽样尽可能覆盖更多失败样例，并在估算中平衡不均匀抽样。

初步实验所用的数据集是 MNIST [16]，这是一个著名的手写数字分类数据集。具体而言，训练使用 1，000 个示例，验证集使用 500 个示例，另有 1，000 个未标记输入（每个周期）作为运行数据集。

DNN-OS 不变量的获取方式如下：

- 领域 不变量：由领域专家根据深度神经网络的输入源定义。具体而言，对于 MNIST，我们假设用户以三种不同形式输入数据，分别要求输入不含直线的数字 {0, 3, 6, 8, 9}、仅含直线的数字 {1, 4, 7}，以及其余数字 {2, 5}。我们为每种形式定义一个不变量。将深度神经网络对每个运行输入给出的输出与该输入源形式所预期的可能数字集合进行比对。

- 数据 不变量：以决策规则的形式从训练数据中自动提取（C4.5 算法 [29]），并根据其置信度（$C \geq 0.99$）和支持度（$S \geq 10$）进行筛选。

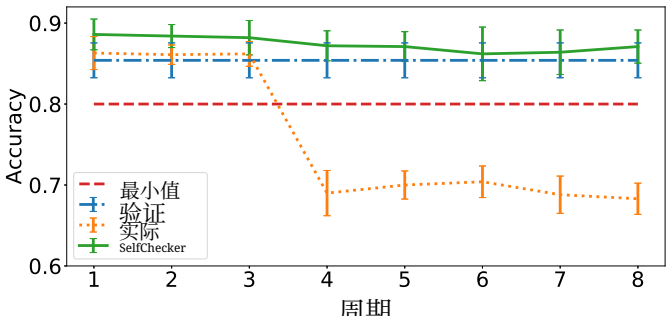

图 2. 使用 SelfChecker 伪预言机得到的 DAIC 结果

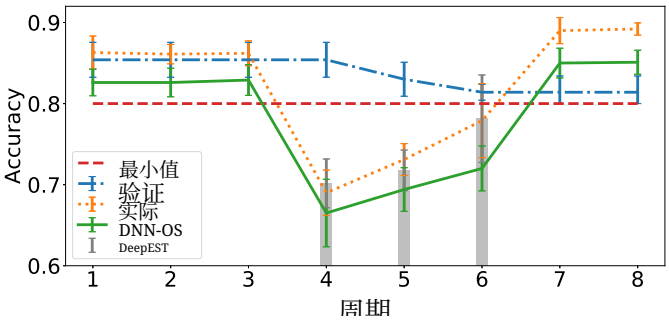

图 3. 使用 DNN-OS伪预言机的 DAIC 结果

表 I 使用 SelfChecker 伪预言机的 DAIC 详细结果

周期	验证准确率	实际准确率	预测准确率 (SelfChecker)	估计准确率 (DeepEST)
1	0.861	0.859	0.881	未触发
2	0.861	0.863	0.883	未触发
3	0.861	0.860	0.879	未触发
4	0.861	0.700	0.868	未触发
5	0.861	0.698	0.861	未触发
6	0.861	0.707	0.866	未触发
7	0.861	0.695	0.864	未触发
8	0.861	0.692	0.867	未触发

表 II 使用 DNN-OS伪预言机的 DAIC 详细结果

周期	验证准确率	实际准确率	预测准确率 (DNN-OS)	估计准确率 (DeepEST)
1	0.861	0.859	0.833	未触发
2	0.861	0.863	0.828	未触发
3	0.861	0.860	0.825	未触发
4	0.861	0.700	0.678	0.713
5	0.833	0.726	0.693	0.719
6	0.810	0.781	0.725	0.784
7	0.818	0.888	0.850	未触发
8	0.818	0.894	0.851	未触发

- 模型 不变量：使用随机森林从验证数据中提取，并将最后一层神经元的输出作为特征。

DeepEST 的样本量为 500，随机抽样示例与加权抽样示例的比例设为 0.5。

如需独立验证或复现，实验代码可在 GitHub 上获取：

https://github.com/dessertlab/DAIC.git.

B. 结果

通过运行八个周期对 DAIC 进行实验，每个周期重复五次。每次迭代都会执行伪预言机评估。

在实验中，基于抽样的评估的触发条件为：

{预测准确率 ＜ （验证准确率 -0.05）} 或
{预测准确率 ＜ 最低准确率}

也就是说，当在线估计的准确率（离线）（预测准确率）与发布前估计的准确率（验证准确率）之间的差值低于给定阈值（此处设为 0.05），或者当预测准确率低于该深度神经网络所要求的最低准确率（实验中设为 0.80）时，就会触发离线评估。

触发 DeepEST 后，一组新的人工标注样本会被发送到数据预处理阶段，并整合到训练集和验证集中。训练数据集中新旧样本的比例可根据上一个或多个周期的准确率估计值进行调整。默认情况下，新旧样本都会纳入考虑。

图2和图3展示了8次DAIC迭代、每次重复5次所得的平均结果和置信区间。表I和表II列出了各个周期的详细信息。前三个周期代表标称条件，即训练集和验证集能够忠实地代表运行数据集。正如预期，使用SelfChecker计算的运行准确率（图2和表I的前三行）以及使用DNN-OS计算的运行准确率（图3和表II的前三行）在前三个周期中均未触发抽样。

从第四个周期开始，通过交换运行数据中的标签2和7来模拟标签偏移。由于SelfChecker仅依赖模型和数据信息，因此无法检测故障，其准确率估计值与实际准确率逐渐偏离，且在剩余周期中未触发通过抽样进行的评估（图2和表I中的周期4）。

得益于领域不变量，DNN-OS能够正确检测故障，并准确估计运行准确率。事实上，当准确率下降时（图3），它会触发通过抽样进行的评估（表II中的周期4）。DeepEST的估计结果证实，运行数据集开始偏离前几个周期中观察到的数据集。因此，新的样本被加入训练数据集，并对模型重新训练。在周期5和6中，DNN-OS仍能正确检测到准确率下降，并利用DeepEST提供的新样本进一步采取改进措施。

在 7th 周期中，模型仅使用在第 4、5 和 6 个周期收集的 1，500 个示例进行训练，从而获得了

较高的实际准确率。在第 7 和第 8 个周期中，DNN-OS 均正确估计出准确率高于 0.8，从而避免触发 DeepEST。

DNN-OS 在第 6 个循环（第 2 次重复）中出现了一次不必要的触发，因为它没有发现实际准确率已经高于最小值。

V. 未来计划

在持续交付场景中采用的基于深度神经网络的系统，其生命周期本质上是迭代的。所提出的伪预言机与抽样技术相结合的方法，既支持对 DNN 准确率的评估，也有助于提升其准确率。该方法帮助工程师利用在运行环境中收集的特征，更忠实地评估深度神经网络，并使其适配所需执行的任务及其具体要求。

我们计划进一步完善 DAIC，制定更复杂的策略，以自动改进闭环中的深度神经网络。DeepEST 等技术能够发现大量失败样例，这些样例与运行特征相结合，有助于提升深度神经网络在边缘情况下的性能。下一步是将自动改进整合到实验阶段和部署阶段。正如初步结果所示，遇到标签偏移等意外现象时，通常很少需要更改性能良好的模型。通常，只需通过将运行样本纳入训练集进行额外训练或从头训练，就足以提高运行准确率。因此，与机器学习运维的理念一致，深度神经网络在线自动改进策略可以基于伪预言机的“概率性”输出。此外，通过自动化数据预处理，无需人工干预即可执行离线再训练步骤。

其次，我们计划将推理引擎应用于运行特征，以自动提取运行约束，从而提高伪预言机在运行期间估计准确率的有效性。Google近期的一项工作强调了将领域知识作为一组规则纳入其中的重要性，以提高机器学习组件的准确率 [30]。

深度神经网络准确率的迭代评估与改进，其意义可能超越经过实验的图像分类领域。我们计划将 DAIC 应用于与工业相关的领域，例如自动驾驶：例如，对油门、制动和转向角进行预测，这些都属于回归问题。

VI. 结论

初步结果证实，在发布前计算的准确率可能与深度神经网络在运行期间、存在标签偏移等意外现象时达到的准确率存在很大差异。然而，得益于领域 不变量，DNN-OS 预测的准确率会随着运行数据中的实际准确率变化，仅在需要时通过抽样触发评估。

DeepEST 提供的估计值可用于可靠地评估运行中的准确率。实验结果还表明，深度神经网络的运行性能可借助新增的有标签样本得到显著提升。

致谢

本项目获得欧盟地平线2020研究与创新计划资助，资助依据为玛丽·斯克沃多夫斯卡-居里资助协议（编号871342“uDEVOPS”）。本项目还获得DIETI COSMIC项目的支持。

参考文献

[1] N. Kühl, M. Goutier, L. Baier, C. Wolff, 和 D. Martin, 《人类与监督机器学习：谁学习模式更快？》*CoRR*，第 abs/2012.03661 卷，2020 年。 [在线]。网址：https://arxiv.org/abs/2012.03661 。K. He、X. Zhang、S. Ren 和 J. Sun, 《深入探究整流器：在[2] ImageNet分类上超越人类水平的表现》，载于国际计算机视觉会议 *(ICCV)*。IEEE，2015 年，第 1026–1034 页。[3] D. Silver、J. Schrittwieser、K. Simonyan、I. Antonoglou、A. Huang、A. Guez、T. Hubert、L. Baker、M. Lai、A. Bolton、Y. Chen、T. Lillicrap、F. Hui、L. Sifre、G. Driessche、T. Graepel 和 D. Hassabis, 《无需人类知识掌握围棋》，《自然》，第 550 卷，第 354–359 页，2017 年 10 月。[4] X. Xie、L. Ma、F. Juefei-Xu、M. Xue、H. Chen、Y. Liu、J. Zhao、B. Li、J. Yin 和 S. See, 《DeepHunter：面向深度神经网络的覆盖率引导模糊测试框架》，载于第 *28* 届 *ACM SIGSOFT* 软件测试与分析国际研讨会 *(ISSTA)*。ACM，2019 年，第 146–157 页。[5] L. Ma、F. Zhang、J. Sun、M. Xue、B. Li、F. Juefei-Xu、C. Xie、L. Li、Y. Liu、J. Zhao 和 Y. Wang, 《DeepMutation：深度学习系统的变异测试》，载于第 *29* 届国际软件可靠性工程研讨会 *(ISSRE)*。IEEE，2018 年，第 100–111 页。[6] Y. Li、W. Shen、T. Wu、L. Chen、D. Wu、Y. Zhou 和 B. Xu, 《高阶变异体测试在深度学习模型上的表现如何：基于二阶变异体分类元组改进测试有效性与效率的细粒度评价》，《信息与软件技术》，第 150 卷，2022 年。[7] B. Recht、R. Roelofs、L. Schmidt 和 V. Shankar, 《ImageNet分类器能否泛化到 ImageNet？》，载于机器学习研究论文集 *(PMLR)*，K. Chaudhuri 和 R. Salakhutdinov 编，第 97 卷，2019 年，第 5389–5400 页。[8] S. Garg、Y. Wu、S. Balakrishnan 和 Z. C. Lipton, 《标签偏移估计的统一视角》，载于第 *34* 届神经信息处理系统会议 *(NeurIPS)*，2020 年。[9] S. Alla 和 S. K. Adari, 《什么是机器学习运维？》，载于《借助 *MLFlow* 入门机器学习运维：在 *AWS SageMaker*、*Google Cloud* 和 *Microsoft Azure* 上部署模型》。加利福尼亚州伯克利：Apress，2021 年，第 79–124 页。[10] Google, 《机器学习中的持续交付与自动化流水线》，网址：https://cloud.google.com/solutions/machine-learning/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning。 [在线]。[11] R. Ashmore、R. Calinescu 和 C. Paterson, 《保障机器学习生命周期：目标、方法与挑战》，《*ACM*计算调查》，第 54 卷，第 5 期，2021 年。[12] E. T. Barr、M. Harman、P. McMinn、M. Shahbaz 和 S. Yoo, 《软件测试中的预言机问题：综述》，《*IEEE*软件工程汇刊》，第 41 卷，第 5 期，第 507–525 页，2015 年。[13] C. Murphy、G. Kaiser、L. Hu 和 L. Wu, 《机器学习应用在蜕变测试中的性质》，载于第 *20* 届软件工程与知识工程国际会议 *(SEKE)*，2008 年，第 867–872 页。[14] A. Guerriero, 《机器学习系统的可靠性评价：预言机问题》，载于*IEEE*国际软件可靠性工程研讨会 *(ISSREW)*，2020 年，第 127–130 页。[15] J. Zhang、M. Harman、L. Ma 和 Y. Liu, 《机器学习测试：综述、研究现状与未来方向》，《*IEEE*软件工程汇刊》，第 48 卷，第 1 期，第 1–36 页，2022 年。[16] Y. LeCun 和 C. Cortes, 《MNIST手写数字数据库》，http://yann.lecun.com/exdb/mnist/，2010 年。

[17] S. Srisakaokul、Z. Wu、A. Astorga、O. Alebiosu 和 T. Xie，《监督学习软件的多实现测试》，载于*AAAI*研讨会。人工智能促进协会，2018 年。[18] K. Pei、Y. Cao、J. Yang 和 S. Jana，《DeepXplore：深度学习系统的自动化白盒测试》，《*ACM*通信》，第 62 卷，第 11 期，第 137–145 页，2019 年。[19] H. Wang、J. Xu、C. Xu、X. Ma 和 J. Lu，《Dissector：通过跨层剖析对深度学习应用进行输入验证》，载于第 *42* 届国际软件工程会议 *(ICSE)*。ACM，2020 年，第 727–738 页。[20] C. Corbière、N. Thome、A. Bar-Hen、M. Cord 和 P. Pérez，《通过学习模型置信度解决故障预测问题》，载于《神经信息处理系统进展》，H. Wallach、H. Larochelle、A. Beygelzimer、F. d'Alché-Buc、E. Fox 和 R. Garnett 编，第 32 卷。Curran Associates, Inc.，2019 年。[21] A. Stocco、M. Weiss、M. Calzana 和 P. Tonella，《自动驾驶系统的异常行为预测》，载于第 *42* 届国际软件工程会议 *(ICSE)*。ACM，2020 年，第 359–371 页。[22] Y. Xiao、I. Beschastnikh、D. S. Rosenblum、C. Sun、S. G. Elbaum、Y. Lin 和 J. S. Dong，《部署中的自检深度神经网络》，载于第 *43* 届国际软件工程会议 *(ICSE)*。IEEE，2021 年，第 372–384 页。[23] J. Quiñonero-Candela、M. Sugiyama、A. Schwaighofer 和 N. D. Lawrence，《训练集与测试集不同时：学习迁移的特征分析》。麻省理工学院出版社，2009 年，第 3–28 页。

[24] A. Guerriero、M. R. Lyu、R. Pietrantuono 和 S. Russo，《使用预言机代理评估基于CNN的图像分类器的运行准确率》，《智能系统与应用》，第17卷，第13期，2023年，第1页–。[25] Z. Li、X. Ma、C. Xu、C. Cao、J. Xu 和 J. Lü，《通过条件化提升运行时DNN测试效率》，载于第*27*届*ACM*欧洲软件工程会议与软件工程基础研讨会联合会议（*ESEC/FSE*）论文集。ACM，2019年，第499–509页。[26] J. Chen、Z. Wu、Z. Wang、H. You、L. Zhang 和 M. Yan，《高效深度神经网络测试的实用准确率估计》，《*ACM*软件工程与方法学汇刊》，第29卷，第4期，2020年。[27] X. Gao、Y. Feng、Y. Yin、Z. Liu、Z. Chen 和 B. Xu，《深度神经网络的自适应测试选择》，载于第*44*届国际软件工程会议（*ICSE*）。ACM，2022年，第73–85页。[28] A. Guerriero、R. Pietrantuono 和 S. Russo，《运行是最严苛的老师：通过寻找错误预测来估计DNN准确率》，载于第*43*届国际软件工程会议（*ICSE*）。IEEE，2021年，第348–358页。[29] J. R. Quinlan，《*C4.5*：机器学习程序》。美国加利福尼亚州旧金山：Morgan Kaufmann出版社，1993年。[30] A. Choudhary，《Google人工智能研究人员推出训练模型的新方法“DeepCTRL”》，《印度分析杂志》（在线），2022年2月13日，https://analyticsindiamag.com/google-ai-researchers-present-a-new-method-to-train-models-deepctrl/，访问日期：2022年10月3日。