arXiv:2305.00735v1 [cs.LG] 1 May 2023

真实世界数据上的无监督异常检测算法：我们需要多少种?

Roel Bouman ROEL.BOUMAN@RU.NL
拉德堡德大学计算与信息科学研究所，*Toernooiveld 212，6525 EC Nijmegen*，荷兰

Zaharah Bukhsh Z.BUKHSH@TUE.NL
埃因霍温理工大学工业工程与创新科学系信息系统专业，*Groene Loper 3，5612 AE Eindhoven*，荷兰

Tom Heskes TOM.HESKES@RU.NL
拉德堡德大学计算与信息科学研究所，*Toernooiveld 212，6525 EC* 奈梅亨，荷兰

摘要

本研究在 52 个真实世界的多变量表格数据集上评估了 32 种无监督异常检测算法，是迄今为止规模最大的无监督异常检测算法比较研究。在这批数据集中，k近邻（到第 k 个最近邻的距离）算法的表现显著优于大多数其他算法。我们对所有数据集中所考察算法的相对表现进行可视化并聚类后，识别出两个明显的簇：一类是“局部”数据集，另一类是“全局”数据集。与邻近样本相比，“局部”异常位于低密度区域，而“全局”异常则位于特征空间中整体密度较低的区域。在局部数据集上，k近邻（第 k 个最近邻）算法表现最佳。在全局数据集上，EIF（扩展隔离森林）算法表现最佳。综合考虑算法的计算复杂度后，我们发现，只需使用包含这三种无监督异常检测算法的工具箱，就足以在这批具有代表性的多变量数据集中发现异常。通过提供代码和数据集，我们的研究很容易复现，也便于扩展更多算法和／或数据集。

Ke关键词： 无监督异常检测、异常分析、算法比较

1. 引言

异常检测研究的是找出不符合数据预期结构的数据点。异常可能是由生成数据的意外过程引起的。在化学领域，异常可能是由实验操作不当造成的；在医学领域，某种疾病可能引发罕见症状；而在预测性维护中，异常可能表明系统即将发生早期故障。根据应用领域的不同，异常具有不同的特性，也可能被称为不同的名称。在机器学习领域（因此本文中也是如此），异常检测通常与离群点检测互换使用。

©2023 Bouman、Bukhsh 和 Heskes。许可协议：知识共享署名 4.0 国际许可协议，参见 https://creativecommons.org/licenses/by/4.0/。

无监督、数据驱动的异常检测是机器学习中的一种标准技术。多年来，人们开发了许多方法或算法来检测异常。其中一些算法旨在解决特定问题，例如高维性。其他方法则尝试从一般意义上检测异常，并侧重于高性能或较低的计算和内存复杂度。由于可用算法众多，很难确定哪种算法最适合特定用例，尤其是对于不熟悉异常检测领域的用户而言。

已有多项研究旨在提供指南，说明何时应使用哪种算法。一些综述研究（Malik et al., 2014；Ruff et al., 2021）根据算法的理论特性提供建议。近年来，也有多项研究在一系列数据集上对多种异常检测算法进行了实证比较。

Emmott 等人（2015）在 19 个数据集上研究了 8 种知名算法。他们发现，孤立森林的总体表现最佳，但建议当数据中存在多个簇时使用 ABOD（基于角度的异常检测）或 LOF（局部异常因子）。

Campos 等人（2016年）在 11 个基础数据集上比较了基于 $12\ k$ 近邻的算法。他们发现，LOF 的表现显著优于许多其他方法，而 KDEOS（核密度估计异常分数）的表现则显著逊于大多数算法。

Goldstein 和 Uchida（2016）在 10 个数据集上比较了 19 种算法。与 Campos 等人不同，Goldstein 和 Uchida 并未进行明确的优化或选择，而是评估了不同合理超参数设置范围内的平均性能。鉴于基于 k 近邻的方法通常能够产生稳定的结果，Goldstein 和 Uchida 推荐使用 kNN（k 近邻）检测全局异常，使用 LOF 检测局部异常，并普遍推荐使用 HBOS（基于直方图的异常选择算法）（关于全局异常和局部异常的解释，见 2.1 节）。Goldstein 和 Uchida 以数据集为单位进行比较，但未进行任何整体统计分析。

最近，Domingues 等人（2018年）在 15 个数据集上应用了 14 种算法，其中一些数据集是分类数据集。他们发现孤立森林（孤立森林）和稳健核密度估计（核密度估计）的表现最佳，但指出对于较大的数据集，稳健核密度估计的计算成本往往过高。

Steinbuss 和 B¨ohm（2021年）提出了一种新颖策略，以若干统计分布作为采样基础，在真实世界数据集中合成异常。他们在多个数据集上比较了 4 种算法，这些数据集由 19 个基础数据集衍生而来，并分别使用原始异常和合成异常进行比较。他们发现，kNN 和孤立森林最擅长检测全局异常，而 LOF 最擅长检测局部异常和依赖异常。同年，Soenen 等人（2021年）研究了超参数优化策略对评估的影响，并提出在小型验证集上优化超参数，再在规模大得多的测试集上进行评估。在对 16 个数据集上的 6 种算法进行比较时，孤立森林表现最佳，CBLOF/u-CBLOF（（非加权）基于簇的局部离群因子）和 kNN 紧随其后；除非使用规模远大于其他算法所用验证集的验证集进行优化，否则单类支持向量机表现最差。

Han 等人（2022）对异常检测方法进行了广泛比较，其中包括监督算法和半监督算法。他们使用开箱即用的超参数设置，在 47 个表格数据集上比较了 14 种无监督算法。他们将较大的数据集下采样至最多 10.000 个样本，并为样本数少于 1000 的数据集重复采样。他们发现无监督算法之间没有显著差异。虽然使用了真实世界数据集，但每个数据集中的异常都是根据 4 种不同的类型定义（见第 2.1 节）合成生成的，并且他们比较了各类型下的表现。此外，他们还通过基于神经网络的特征提取，分析了 CV 和 NLP 中使用的更复杂基准数据集，例如 CIFAR10（Krizhevsky 等人，2009）和亚马逊数据集（He 和 McAuley，2016）。

其他研究的范围较为有限，例如只涵盖针对高维数据的方法（Xu 等，2018年），或只考虑集成方法（Zimek 等，2014年）。

Campos 等人（2016年）、Goldstein 和 Uchida（2016年）、Domingues 等人（2018年）、Steinbuss 和 B¨ohm（2021年）、Soenen 等人（2021年）以及 Han 等人（2022年）的研究在用作基准测试时存在若干局限。首先，除 Han 等人的研究外，其他研究使用的数据集数量都相当少。其次，这些研究涵盖的方法数量有限。Campos 等人只比较基于 kNN 的方法，而 Goldstein 和 Uchida 未能涵盖近几年日益流行的许多方法，例如孤立森林（Liu 等人，2008年）及其变体（Hariri 等人，2019年）。Soenen 等人只考察了 6 种常用方法，Steinbuss 和 B¨ohm 涵盖了 4 种方法，而 Han 等人则涵盖了 14 种无监督方法。

其中一些研究考察了包含特定类型异常的数据集上的性能，例如全局异常或局部异常。具体而言，Steinbuss 和 B¨ohm 考察了不同算法在包含合成全局异常、局部异常和依赖异常的数据集上的性能。同样，Han 等人合成了这三类异常以及簇异常，用于其比较研究。据我们所知，Goldstein 和 Uchida 的研究是唯一分析现实世界中的全局异常和局部异常，即非合成异常的研究。具体来说，他们分析了“pen-local”和“pen-global”数据集，这两个数据集是同一数据集的两个变体，其中选择了不同的类别，以专门获得局部异常和全局异常。

在实际应用中，我们对常用基准数据集中存在的异常类型知之甚少，因此，针对特定异常类型在真实数据上开展的大规模比较仍然缺失。本研究在大量多变量数据集上应用了众多常用异常检测方法，以探寻何时应采用何种算法的指导原则。我们明确选择不进行超参数优化，以便以真正无监督的方式评估算法的性能。相反，我们在一系列合理的超参数设置下评估每种算法，并比较其平均性能。这与 Soenen 等人（2021年）的做法不同；他们在一个小型验证集上进行广泛优化，从而为半监督检测或主动学习提供指导。我们的方法则更接近 Domingues 等人（2018年）采用的做法，他们同样比较开箱即用性能。据我们所知，这是迄今为止同类研究中规模最大的一项。

2. 背景

2.1 异常类型

关于异常及其属性，存在许多不同的定义，其中许多是在彼此孤立的背景下提出的，没有考虑它们与其他定义或属性之间的关系（Breunig 等人，2000年）。近来，Foorthuis（2021年）的一篇综述试图统一异常检测多个子领域中的定义，以涵盖所有类型的异常。然而，这些定义并未涵盖许多异常属性或类型，例如聚类异常或依赖异常。

我们并不试图重新定义每一种不同的异常类型，而是将异常视为可能具有多个有时并不互斥的属性。相反，我们定义了四个尺度的非互斥属性；这些属性组合起来，涵盖了我们所知文献中多变量表格数据中的所有异常类型。

2.1.1 Enclosed and peripheral anomalies

在特征空间中，异常可能被正常数据包围。当这种情况发生时，我们将其定义为包围型异常。在该轴的另一端，边缘型异常占据特征空间的边缘，并且其一个或多个属性得分低于正常数据区域得分的最小值，或高于其最大值。包围型异常和边缘型异常的示例分别见图 1a 和图 1b。

2.1.2 Global and local density anomalies

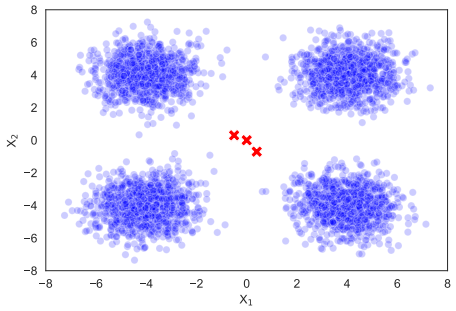
最常讨论的异常类型是全局异常和局部异常。这些定义源自 Breunig 等人（2000）的研究。全局异常是能够与正常数据区分开的点，因为它们位于特征空间中密度较低的区域。然而，仅使用这些标准无法分离局部异常。局部异常位于相对于附近正常区域密度较低的区域中。即使特征空间中存在多个具有不同密度函数的簇，这也能识别出定义明确的异常。全局密度异常和局部密度异常的示例分别见图 1c 和图 1d。

2.1.3 Isolated and clustered anomalies

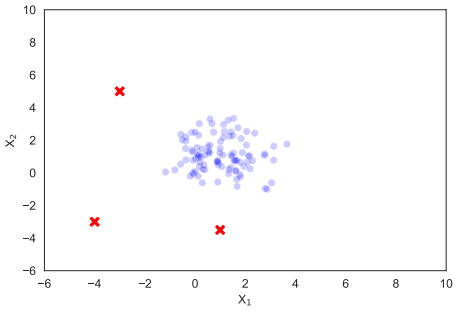
异常通常是孤立的，由单个数据点构成，附近没有其他正常或异常数据点。在许多实际情况下，异常并非如此孤立，少量异常会形成簇，从而构成聚类异常。聚类异常与一种称为“掩蔽”的现象密切相关：相似的异常会通过形成簇来掩盖彼此的存在（Liu 等人，2008）。图 1e 和图 1f 分别展示了孤立异常和聚类异常的示例。

2.1.4 Univariate and multivariate anomalies

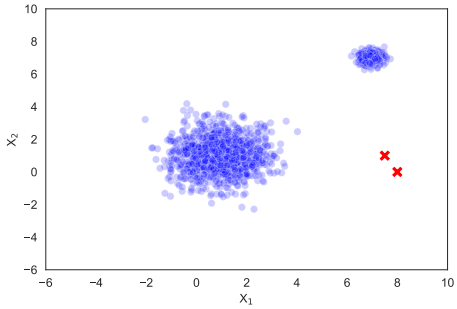
有些异常显然属于单变量异常。也就是说，只需一个处于异常范围内的特征得分，就能识别出这些异常。另一些异常则属于多变量异常，需要特征得分的特定组合才能将其识别为异常。这些



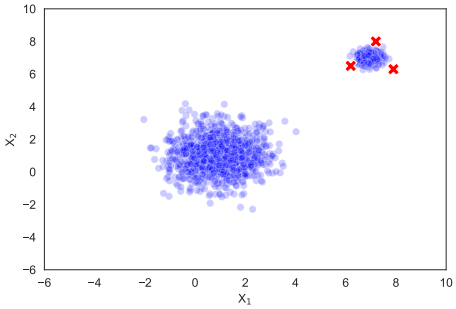
（a）封闭异常示例



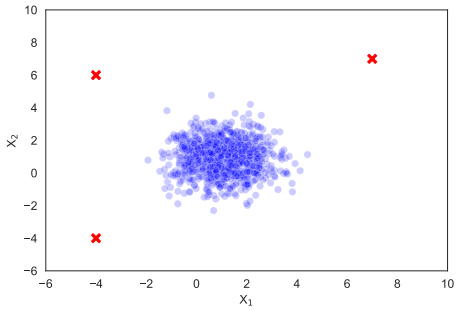
（b）外围异常示例



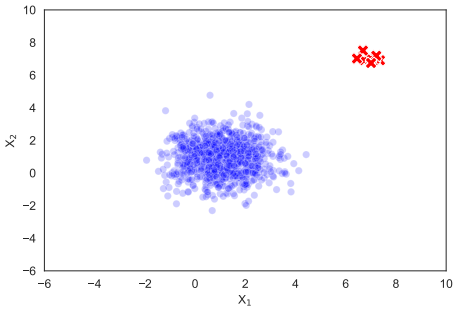
（c）全局异常示例



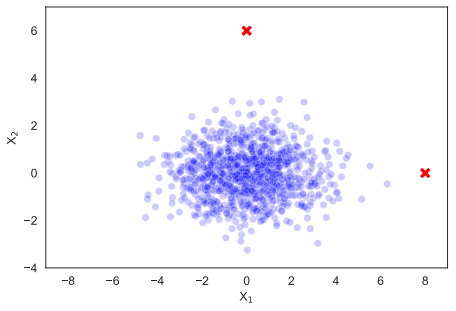
（d）局部异常示例



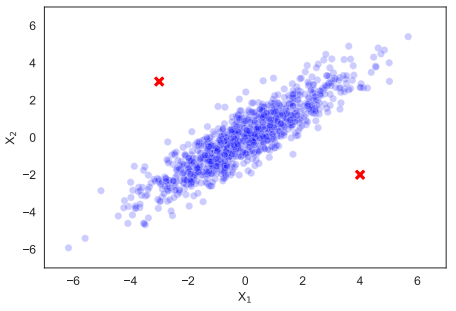
（e）孤立异常示例



（f）聚类异常示例



(g) 单变量异常示例



(h) 多变量异常示例

图1：沿4个已定义的属性轴展示的8个不同类型的异常示例。正常数据以蓝色点表示，异常以红色叉号表示。

多变量异常也常被称为依赖异常，因为它们不同于数据的正常依赖结构或因果结构。孤立异常和聚类异常的示例分别见图 1g 和图 1h。

3. 材料与方法

我们评估了表 1 所列的 32 种不同算法的有效性。对于每个数据集，我们多次评估每种算法，每次使用一组不同且合理的超参数。随后对不同超参数设置下的结果取平均值，从而为每种方法与数据集的组合得到一个平均 ROC-AUC 分数。我们不进行超参数优化，例如使用交叉验证，以反映现实世界中的情况：没有标签可用于训练模型。尽管 Thomas 等人（2016年）研究过无监督超参数优化，但该方法尚未应用于本研究所考虑的大多数算法。

3.1 算法

在这 32 种方法中，有 27 种采用了用于异常检测的热门 Python 库 PyOD（Zhao 等人，2019年）中的实现。作为本研究的一部分，我们为这一开源库做出了多项贡献，例如对 COF（基于连通性的离群因子）方法进行了内存高效的实现，并实现了 Birg´e 和 Rozenholc（2006年）提出的直方图大小选择方法，该方法用于 HBOS 和 LODA（轻量级在线异常检测器）。对于 EIF（扩展隔离森林），我们使用了作者在 Python 软件包“eif”中提供的实现，该软件包由 Hariri 等人（2019年）开发。我们用 Python 实现了 ODIN（使用入度数进行离群检测）方法，目前正准备将其提交至 PyOD 软件包。集成LOF方法使用 PyOD 中的基础 LOF 算法实现。DeepSVDD 基于其作者 Lukas Ruff[1]，公开提供的代码应用，我们对其进行了修改，使其能够处理通用表格数据集。

我们省略了 PyOD 软件包中实现的若干方法，例如 LOCI 和 ROD，因为它们的时间或内存复杂度为 $\mathcal{O}(n^3)$，其中 n 表示数据点的数量。由于 PyOD 的 SOS 方法具有 $\mathcal{O}(n^2)$ 的内存需求，我们也忽略了该方法。在较小的数据集上进行评估时，这些方法的表现均不如我们纳入的其他算法，且没有任何特别出色之处。

3.2 数据

3.2.1 Datasets

在本研究中，我们考察了来自各种来源的大量数据集。我们重点关注实值、多变量的表格数据，这些数据与 Fern´andez-Delgado 等人（2014）、Campos 等人（2016）、Goldstein 和 Uchida（2016）、Soenen 等人（2021）以及 Domingues 等人（2018）所使用的数据集具有可比性。表2汇总了这些数据集的信息，列出了每个数据集的来源、样本数量、变量数量、异常值数量及其占比，以及在数据处理过程中删除的重复项数量。

1. https://github.com/lukasruff/Deep-SVDD-PyTorch

表 1：算法概述、超参数设置、首次发表年份及作者。对于神经网络，“收缩因子”超参数表示编码器中的任何后续层均按以下方式定义：层大小$_{n+1}$ = 层大小$_n$ ×收缩因子。

Name	超参数	出版
ABOD	FastABOD，$k = 60$	Kriegel 等人（2008 年）
AE	$n_{\text{layers}} = 1$，2，3，收缩因子= 0.2，0。3，0。5	Japkowicz 等人（1995年）
ALAD	$n_{\text{layers}} = 3$，收缩因子= 0.2，0。3，0。5	Zenati 等人（2018年）
CBLOF	$k = 2$，..，14，$\alpha = 0.7$，0.8，0.9，$\beta = 3$，5，7	He 等人（2003年）
COF	$k = 5$,10,15,20,25,30	Tang 等人（2002年）
COPOD		Li 等人（2020）
DeepSVDD	$n_{\text{layers}} = 1, 2, 3$，收缩因子= 0.2,0.3, 0.5	Ruff 等人（2019）
ECOD		Li 等人（2022）
EIF	$n_{\text{trees}} = 1000$, $n_{\text{samples}} = 128$，256，512，1024，无放回，扩展层级：1，2，3	Hariri 等人（2019）
集成LOF	$k = 5$上的最大 LOF 分数，.....30	Breunig 等人（2000年）
gen2out		Lee 等（2021）
GMM	$n_{\text{gaussians}} = 1, ..., 15$	Agarwal（2007）
HBOS	n_{bins} 基于 Birg´e-Rozenholc 准则	Goldstein 和 Dengel（2012）
IF	$n_{\text{trees}} = 1000$, $n_{\text{samples}} = 128$，256，512，1024，无放回	Liu 等人（2008）
INNE	200 个估计器	Bandaragoda 等人（2018年）
KDE	高斯核	Latecki 等人（2007）
kNN	$k = 5$，.....30，均值距离	Ramaswamy 等人（2000）
kth-NN	$k = 5$，.....30，最大距离	Ramaswamy 等人（2000）
LMDD	$n_{\text{shuffles}} = 100$，MAD 不相似度函数	Arning 等人（1996）
LODA	n_{bins} 基于 Birg´e-Rozenblac 准则，100 个随机投影	Pevn`y（2016年）
LOF	$k = 5$、.....、30	Breunig 等人（2000年）
LUNAR	$k = 5$，10，15，20，25，30	Goodge等（2022）
MCD	子集比例= 0.6，0。7，0。8，0。9	Rousseeuw和Driessen（1999）
单类支持向量机	径向基函数核，$\nu = 0.5, 0.6, 0.7, 0.8, 0.9$, $\gamma = 1/d$	Sch¨olkopf 等人（1999年）
ODIN	$k = 5$,...30	Hautamaki等（2004）
PCA	选定的主成分解释了 > 30，50，70，90%的方差	Shyu等人（2003）
sb-DeepSVDD	$n_{\text{layers}} = 1$，2，3，收缩因子= 0.2，0。3，0。5	Ruff 等人（2019年）
SOD	$k = 20$，25，30，$l = 10$，14，18，$\alpha = 0.7$，0。8，0。9	Kriegel等人（2009）
SO-GAAL	停止轮数= 50	Liu等人（2019）
u-CBLOF	$k = 2$,.., 14, $\alpha = 0.7, 0.8, 0.9$, $\beta = 3, 5, 7$	Amer和Goldstein（2012）
VAE	$n_{\text{layers}} = 1$，2，3，收缩因子= 0.2，0。3，0。5	An 和 Cho（2015）
β-变分自编码器	$n_{\text{layers}} = 1$，2，3，收缩因子= 0.2，0。3，0。5，$\gamma = 10$，20，50	Zhou 等人（2020）

我们的数据集集合主要由 ODDS 数据集合（Rayana，2016年）中的数据集构成，具体来说，是多维点数据集。这是一个由各种数据集组成的集合，其中大多数改编自 UCI机器学习库（Dua 和 Graff，2017年）。所有数据集均为实值数据，不包含任何分类数据。遗憾的是，该集合的整理并未完全更新，导致其中一些列出的数据集无法获取。无法获取的数据集已从此次比较中省略。

除 ODDS 数据集外，我们还纳入了早期异常检测研究中使用的公开数据集：Goldstein 和 Uchida（2016年）的比较研究中的若干数据集，Campos 等人（2016年）使用 ELKI 进行的比较研究中的若干数据集（Schubert 和 Zimek，2019年），一项关于生成式对抗主动学习（GAAL）的研究中的若干数据集（Liu 等人，2019年），一项关于扩展自编码器的研究中的若干数据集（Shin 和 Kim，2020），ADBench 比较研究中的若干数据集（Han 等人，2022年），以及一项关于高效在线异常检测（EOAD）的研究中的若干数据集（Brandsæter 等人，2019年）。这些后续来源中与 ODDS 集合内已有数据集重复或近似重复的数据集均被排除。在表2中，我们准确说明了每个数据集的下载或重建来源。

Emmott 等（2013，2015）提出了一套系统的方法来构建异常检测基准测试，Pevn`y（2016）也广泛采用了这一方法。本文没有构建自己的基准数据集，因为这不可避免地会带有一定的主观性，并可能产生偏差；我们转而使用先前比较研究中采用的大量不同数据集。本研究不包含合成数据集，因为真实世界数据集通常被认为是用于算法基准测试的最佳现有工具（Emmott 等，2015；Domingues 等，2018；Ruff 等，2021）。尽管基准测试更倾向于使用真实世界数据集，但我们也注意到，在研究异常检测算法的特定属性时，合成数据很有用。

3.2.2 Preprocessing

为比较各种算法在不同数据集上的性能，我们采取了若干步骤。首先，从每个数据集中删除所有重复样本，因为许多异常检测方法无法处理含有重复项的数据（Campos 等，2016）。此外，对所有数据集中所有变量进行缩放和中心化。尽管有些算法（如孤立森林）能够隐式处理尺度不同的变量，但涉及距离或交叉乘积计算的方法会受到变量尺度的显著影响。对数据集中的每个变量，中心化的做法是减去其中位数；缩放则是将每个变量除以其四分位距。我们有意选择这种中心化和缩放流程，因为与均值和标准差相比，中位数和四分位距受异常影响较小。当已知存在异常时，通常认为这一流程比标准化更稳定（Rousseeuw 和 Croux，1993）。

3.3 评估流程

由于异常检测具有无监督的特性，通常评估异常分数比评估某些算法也会生成的二元标签更有用。利用分数可以对样本进行排序，从而深入了解异常的内在特征。

表2：我们在异常检测算法比较中使用的52个多变量数据集概要：数据集的俗称、来源、去除重复项后的样本数、变量数、异常数和移除的重复项数，以及异常所占百分比。

Name	来源	#样本数	#变量数	#离群值	%离群值	#重复项	#已移除的变量
aloi	Goldstein	49533	27	1507	(3.04%)	466	0
annthyroid	ODDS	7062	6	534	(7.56%)	138	0
心律失常	ODDS	452	257	66	(14.6%)	0	17
后门	ADBench	87020	193	1879	(2.16%)	8309	3
breastw	ADBench	449	9	236	(52.56%)	234	0
竞选活动	ADBench	41176	62	4639	(11.27%)	12	0
cardio	ODDS	1822	21	175	(9.6%)	9	0
CelebA	ADRepository	113983	39	2911	(2.55%)	88616	0
覆盖物	ODDS	286048	10	2747	(0.96%)	0	0
捐赠者	ADBench	35421	10	7667	(21.65%)	583905	0
故障	ADBench	1941	27	673	(34.67%)	0	0
欺诈	ADBench	275661	29	473	(0.17%)	9146	0
玻璃	ODDS	213	9	9	(4.23%)	1	0
肝炎	ELKI	80	19	13	(16.25%)	0	0
HRSS 异常优化_	ex-AE	5874	18	1468	(24.99%)	13760	0
HRSS 异常标准_	ex-AE	7040	18	1830	(25.99%)	16605	0
http	ODDS	221900	3	72	(0.03%)	345598	0
互联网广告	ELKI	1966	1555	368	(18.72%)	0	0
电离层	ODDS	350	33	125	(35.71%)	1	0
陆地卫星	ADBench	6435	36	1333	(20.71%)	0	0
信件	ODDS	1598	32	100	(6.26%)	2	0
MAGIC伽马射线数据集	ADBench	18905	10	6573	(34.77%)	115	0
乳腺摄影	ODDS	7848	6	253	(3.22%)	3335	0
mi-f	ex-AE	24945	40	2048	(8.21%)	341	5
mi-v	ex-AE	24945	40	3942	(15.8%)	341	5
MNIST	ODDS	7603	78	700	(9.21%)	0	22
musk	ODDS	3062	166	97	(3.17%)	0	0
nasa	ex-AE	4687	32	755	(16.11%)	0	0
光学数字	ODDS	5198	62	132	(2.54%)	18	2
pageblocks	ELKI	5393	10	510	(9.46%)	0	0
帕金森	ELKI	195	22	147	(75.38%)	0	0
pen-global	Goldstein	808	16	90	(11.14%)	0	0
pen-local	Goldstein	6723	16	10	(0.15%)	0	0
pendigits	ODDS	6870	16	156	(2.27%)	0	0
pima	ODDS	768	8	268	(34.9%)	0	0
卫星	ODDS	6435	36	2036	(31.64%)	0	0
satimage-2	ODDS	5801	36	69	(1.19%)	2	0
seismic-bumps	ODDS	2578	21	170	(6.59%)	6	3
Shuttle	ODDS	49097	9	3511	(7.15%)	0	0
skin	ADBench	51433	3	14654	(28.49%)	193624	0
smtp	ODDS	71230	3	21	(0.03%)	23926	0
垃圾邮件基准数据集	GAAL	4206	57	1678	(39.9%)	0	0
语音	ODDS	3686	400	61	(1.65%)	0	0
stamps	ELKI	340	9	31	(9.12%)	0	0
甲状腺	ODDS	3656	6	93	(2.54%)	116	0
vertebral	ODDS	240	6	30	(12.5%)	0	0
元音	ODDS	1452	12	46	(3.17%)	4	0
波形	GAAL	3442	21	99	(2.88%)	0	0
wbc	ODDS	377	30	20	(5.31%)	1	0
wbc2	ADBench	223	9	10	(4.48%)	0	0
wilt	ELKI	4819	5	257	(5.33%)	0	0
wine	ODDS	129	13	10	(7.75%)	0	0
wpbc	ADBench	198	33	47	(23.74%)	0	0
yeast	ADBench	1453	8	481	(33.1%)	31	0
yeast6	EOAD	1453	8	35	(2.41%)	31	0

对于每个数据集，我们一次性在所有可用数据上计算异常分数，不使用任何交叉验证或训练测试划分。随后结合真实标签，利用无监督分析得到的分数评估算法的性能。为了比较不同算法，我们计算每种算法-数据集组合的性能，指标为 ROC（受试者工作特征）曲线所对应的 AUC（曲线下面积）值。这是异常检测评估中最常用的指标（Goldstein 和 Uchida，2016年；Campos 等，2016年；Xu 等，2018年），可以从概率的角度轻松解读。我们曾考虑使用其他指标，例如 R-精确率或平均精确率，以及 Campos 等人（2016年）提出的经机会校正的变体，但发现这些指标稳定性较差，也更难解读。

对于每个数据集，我们根据各算法生成的分数计算 AUC 分数，并对其进行排序。遵循 Demšar（2006）关于分类器比较的建议，我们使用 Iman-Davenport 统计量（Iman 和 Davenport，1980年）来确定算法之间是否存在显著差异。如果该统计量低于与 p 值 0.05 相对应的预设临界值，我们就会进行 Nemenyi 事后检验（Nemenyi，1963年），以评估哪些算法之间存在显著差异。

在本文的部分可视化图中，我们绘制了最大 AUC 百分比，其定义如下：

$$\widetilde{\mathrm{AUC}}(a,d) = \frac{\mathrm{AUC}(a,d)}{\max_{a' \in A} \mathrm{AUC}(a',d)} \times 100\,,$$

其中， a 一个 A 算法和 d 一个 D 数据集。

3.4 可复现性

为了复现我们的所有实验，我们提供了一个公开的 GitHub 代码仓库[2] ，其中包含所有实验以及本文所有图表制作所用的代码和数据集。

4. 结果

4.1 算法总体性能

为了评估性能，我们使用与 ROC 曲线对应的 AUC 指标，在每个数据集上评估了每种算法。为了在多个合理的超参数下评估性能，某一方法的 AUC 值取所评估的每个超参数设置对应 AUC 的平均值。在分析中，我们发现有三个数据集几乎所有参与评估的算法表现都不佳，即所有 AUC 值均介于 0.4 和 0.6 之间。因此，这些数据集，即“hrss 异常标准”数据集、“wpbc”数据集和“yeast”数据集，被排除在后续分析之外。有些数据集没有高于 0.6 的 AUC 值，但存在低于 0.4 的 AUC 值。在这些情况下，检测器在反转标签后表现更好。在“skin”和“vertebral”数据集中观察到了这种现象。这些数据集的构建基于这样的处理方式：将样本数量最多的组视为正常类（0），将较小的组视为异常类（1）。然而，对于这两个

2. https://github.com/RoelBouman/异常值检测

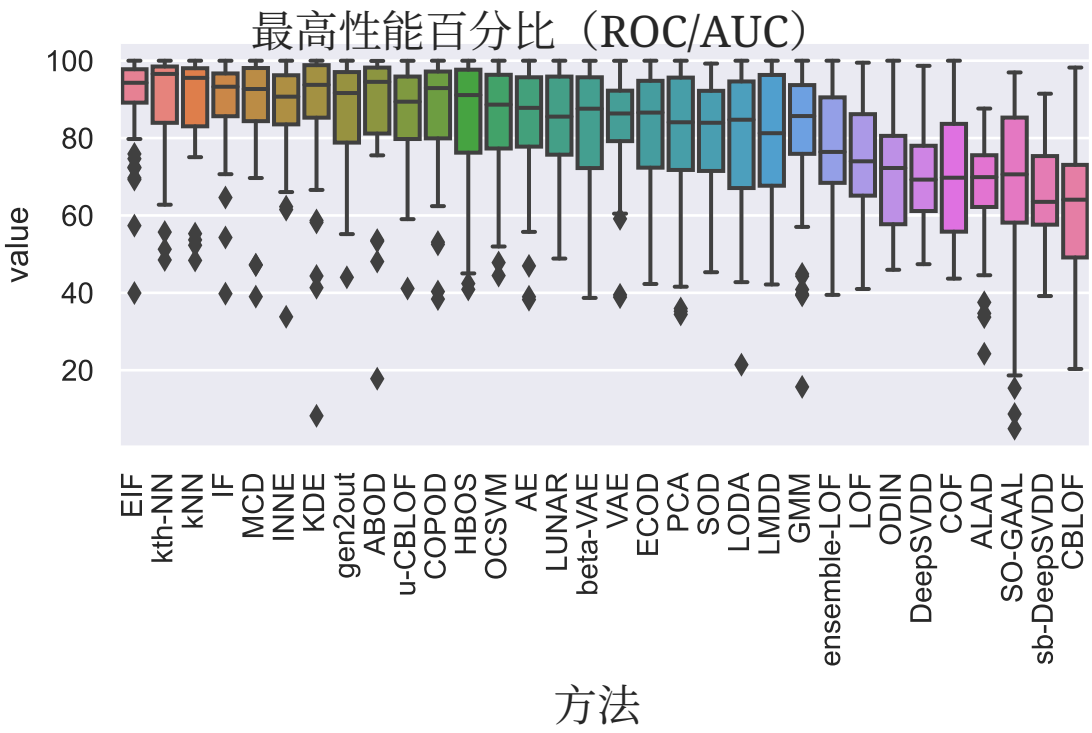


图2：各算法在各数据集上的性能箱线图，以最高 AUC 的百分比表示。最高 AUC 是该数据集上表现最佳的算法所取得的 AUC 最高值。箱线图的须线从下四分位数和上四分位数分别延伸至超出四分位距 1.5 倍的位置。须线范围之外的数据集与算法组合以菱形标示。

数据集，更具异质性的组被选为正常类，这与正常异常定义相矛盾。对于这些数据集，我们反转了标签并重新计算了 AUC 值。

图2展示了各方法性能的分布。为了比较不同数据集上的 AUC（这些数据集的基准性能可能不同），我们在箱线图中以各数据集上表现最佳的算法所取得的最高 AUC 值的百分比来表示 AUC。

可以看出，许多算法的表现相近，最高 AUC 的中位百分比约为 90%。此外，还可以观察到若干较低的中位数以及较宽的四分位数范围。

为确定图2中观察到的性能差异是否具有显著性，我们应用了Iman-Davenport检验。该检验得出的检验统计量为18.076，远高于临界值0.620，因此拒绝了零假设。随后，我们应用Nemenyi 事后检验，以确定哪些算法显著优于其他哪些算法。该Nemenyi 事后检验的结果汇总于表3中。

表3：基于Nemenyi事后分析的算法间显著差异。++/+ 表示行算法分别以 $p = 0.05/p = 0.10$的显著性水平优于列算法，而--表示行算法在 $p = 0.05$的显著性水平上被列算法超越。行和列分别按均值性能降序和升序排列。当列算法在 $p = 0.05$ 或 $p = 0.10$的显著性水平上不被任何其他算法超越时，不显示该列。最后一列显示均值AUC。

	CBLOF	sb-DeepSVDD	SO-GAAL	ALAD	COF	DeepSVDD	ODIN	LOF	ensemble-LOF	LMDD	LODA	SOD	PCA	VAE	beta-VAE	**Mean AUC**
EIF	++	++	++	++	++	++	++	++	++	++	++	++	++			0.773
kth-NN	++	++	++	++	++	++	++	++	++	++	++	++	++	+	++	0.769
kNN	++	++	++	++	++	++	++	++	++	++	++	++	++			0.766
IF	++	++	++	++	++	++	++	++	++		+		++			0.764
MCD	++	++	++	++	++	++	++	++								0.756
INNE	++	++	++	++	++	++	++	++								0.749
KDE	++	++	++	++	++	++	++	++	++	+	++	+	++			0.748
gen2out	++	++	++	++	++	++	++	++								0.748
ABOD	++	++	++	++	++	++	++	++	+							0.747
u-CBLOF	++	++	++	++	++	++	++									0.741
COPOD	++	++	++	++	++	++	++	++								0.738
HBOS	++	++	++	++	++	++	++									0.725
单类支持向量机	++	++	++	++	++	++	++									0.723
AE	++	++	++	++		++	+									0.720
LUNAR	++	++	++	++		++	+									0.711
β变分自编码器	++	++	+	++		++										0.710
VAE	++	++	++	++		++										0.710
ECOD	++	++	++	++		++	+									0.709
PCA		++														0.692
SOD	++	++														0.687
LODA	+	++														0.685
LMDD	++	++														0.684
GMM	++	++	++	++		++	+									0.683
集成LOF		++														0.661
LOF																0.630
ODIN																0.595
DeepSVDD														--	--	0.593
COF																0.592
ALAD														--	--	0.569
SO-GAAL														--	-	0.564
sb-DeepSVDD									--	--	--	--	--	--	--	0.553
CBLOF										--	-	--		--	--	0.519

表3显示，确实有几种算法的性能显著优于许多其他算法。其中最突出的是 kth-NN，在 $p = 0.05/p = 0.10$ 显著性水平下，其性能显著优于本研究评估的14/15种算法。广受欢迎的孤立森林方法及其扩展版本也表现出类似的稳定性能。由于孤立森林及其变体的计算复杂度会随样本数量 n呈线性增长，因此对于大型数据集而言，相较于 kNN及其衍生方法，它们可能具有明显优势；后者的计算复杂度呈平方级增长，或者在优化后至少以 $\mathcal{O}(n\log\ n)$ 增长。

从图2和表3中还可以观察到，原始CBLOF方法的性能远远最差，在 $p = 0.05/p = 0.10$ 显著性水平下，其性能显著逊于22种算法中的21种。这与Goldstein 和 Uchida（2016年）的结果相一致；他们同样发现CBLOF的表现持续欠佳，而其未加权变体u-CBLOF的性能则与其他算法相当。

从这些总体结果可以清楚看出，许多神经网络的表现并不理想。（软边界）DeepSVDD、ALAD 和 SO-GAAL 都处于总体方法性能的较低区间。我们推测，这一现象可能有两个原因。首先，这些方法并非针对表格数据设计，因此无法利用使其在典型计算机视觉任务中占据优势的同等特征提取能力。其次，这些方法相对复杂，很难指定适用于各种数据集的通用超参数设置和架构。并非所有神经网络都存在这一问题：自编码器及其变体，以及 LUNAR 的表现大致处于平均水平。这可能是因为它们采用了更直接的优化准则。

除了神经网络之外，LOF、ODIN、COF 和 CBLOF 等局部方法也是表现最差的方法之一。LOF 的这一结果与 Campos 等人（2016）的研究结果形成鲜明对比；他们发现 LOF 是表现最佳的方法之一。这很可能是因为他们仅在少量异常比例较低的数据集上进行了评估，因此 LOF 受到淹没或掩蔽的影响较小（Liu 等人，2008）。我们将在第4.3节进一步研究这一发现。

4.2 聚类算法和数据集

为了分别直观展示算法之间以及数据集之间的相似性，图3展示了各数据集与算法组合的性能热图，以及两个层次聚类的树状图：一个针对数据集，另一个针对算法。我们采用皮尔逊相关作为距离度量进行平均连接聚类分析，并使用 Bar-Joseph 等人（2001）的方法优化叶节点排序，以便于可视化。

图3显示，许多相似的算法按预期聚集在一起，各算法家族也形成了各自的簇。从更宏观的层面可以观察到一些有趣的模式。对于这些算法，我们得到若干相当明显的簇。首先，CBLOF 独树一帜，因为它在几乎所有数据集上的表现都较差。同样，SO-GAAL、GMM 和 ALAD 等表现欠佳的方法，也无法明确归入其他逐渐形成的簇。局部方法 COF、集成LOF、LOF 和 ODIN 则形成了一个独立的簇。这些专门用于检测局部异常的算法，在……上表现良好

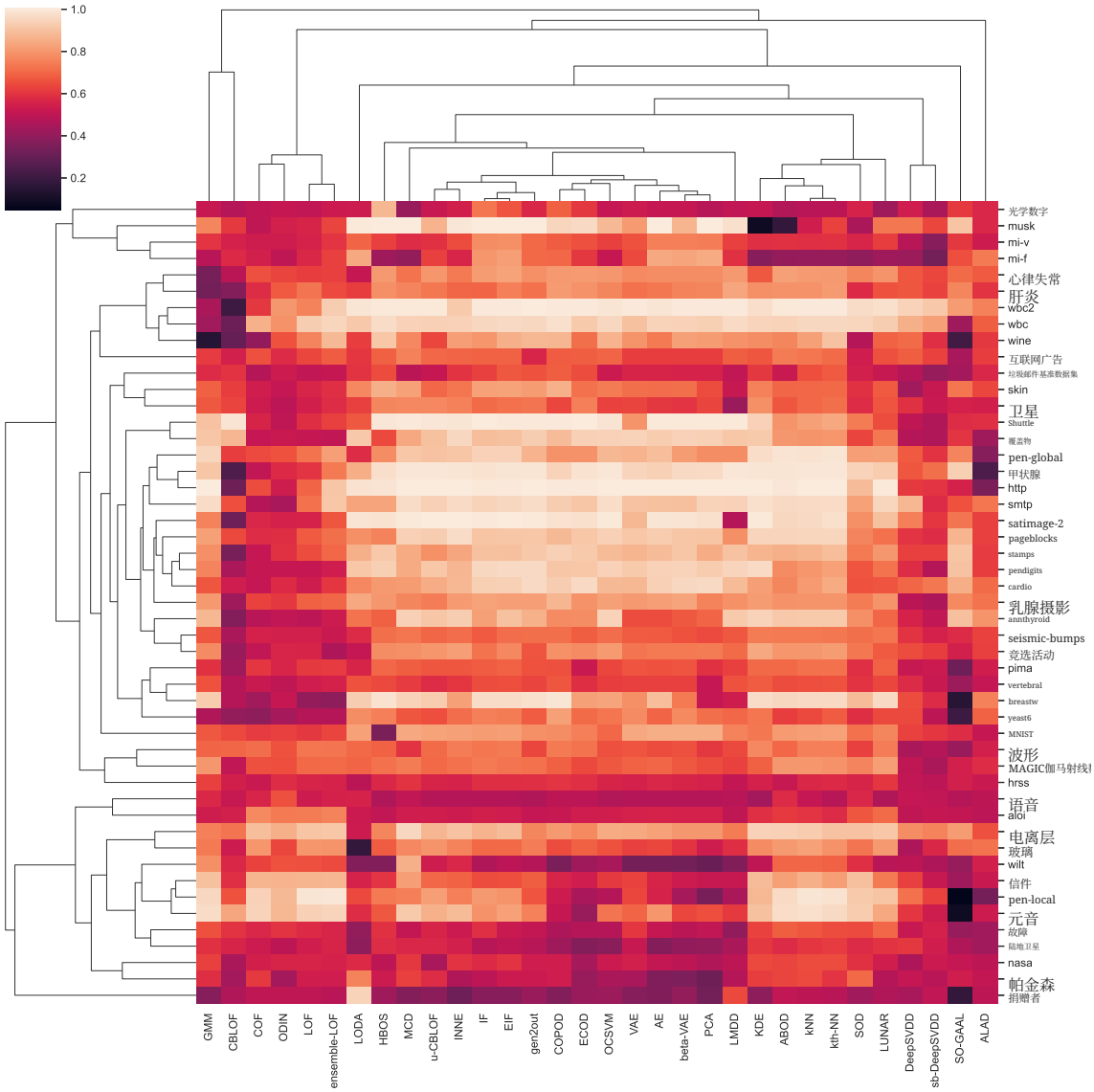


图3：各算法的 ROC/AUC 性能聚类热图。算法和数据集分别采用层次聚类，以平均连接法和皮尔逊相关作为度量。

少数（约四分之一）的数据集，但在大多数其他数据集上的表现不佳。我们发现，kNN 及 ABOD 和 SOD 等相关方法形成了一个小簇，在所有数据集上的表现都还不错。最后，有一个较大的方法簇似乎与局部方法呈负相关：它们在大多数（约四分之三）的数据集上表现良好，但在其余数据集上的表现则不那么出色。

这些数据集明显分成两个不同的簇：一个簇中的数据集适合局部算法，另一个簇中的数据集则适合那个较大的算法簇。结合双向聚类结果和对算法的了解可以推断，约四分之一的数据集属于所谓的局部问题，而其余四分之三属于全局问题。专门构造的局部集和全局集“pen-local”与“pen-global”也印证了这一点，它们显然分别属于预期的簇。
Steinbuss 和 B¨ohm（2021）以及 Emmott 等人（2015）的研究也支持这一观察：他们同样发现了被归类为局部／依赖型异常和多簇异常的异常，与全局异常之间存在差异。通过我们的分析，我们无法清楚地观察到其他由异常属性差异所形成的明显模式。

据我们所知，先前没有研究详细考察过，在真实世界数据中自然出现的异常检测问题上，算法分别用于全局异常检测和局部异常检测时，其性能有何差异。

4.3 全局与局部问题上的性能

在上一节中，我们发现数据集明显分为两簇：一簇是“局部”数据集，包括“aloi”“donors”“fault”“glass”“ionosphere”“landsat”“letter”“nasa”“parkinson”“pen-local”“vowels”和“wilt”；另一簇则是其余的“全局”数据集。考虑到不同方法可能在不同类型的数据集上表现良好，我们针对这两簇分别重复了第4.1节中的显著性检验流程。

所有算法应用于局部数据集集合时的性能箱线图见图4。图4清楚地展示了部分局部异常检测方法的性能反转。在整个数据集集合中表现最差的COF、集成LOF和LOF，在应用于其专门开发所针对的问题时，却跻身表现最佳的方法之列。这一现象是辛普森悖论的典型例子（Simpson，1951年）。这也部分解释了总体比较结果与Campos 等人（2016年）的比较结果之间为何存在差异。

随后，我们仅针对局部数据集重复进行了 Nemeny-Friedman 事后检验。表4总结了这项分析的结果。kNN 和 ABOD 表现最佳，且显著优于另外16种方法，显著性水平为 $p = 0.05$。

随后，我们对全局数据集重复进行了分析，得到了图5中的性能箱线图以及表5中的显著性检验结果。

从图5和表5可以看出，扩展隔离森林的均值性能最高，紧随其后的是标准隔离森林。扩展隔离森林分别在 $p = 0.05/p = 0.10$ 下优于 15 种方法中的 13 种。巧合的是，这些方法的计算需求和内存需求也最低，因此它们最有可能适用于全局异常检测。

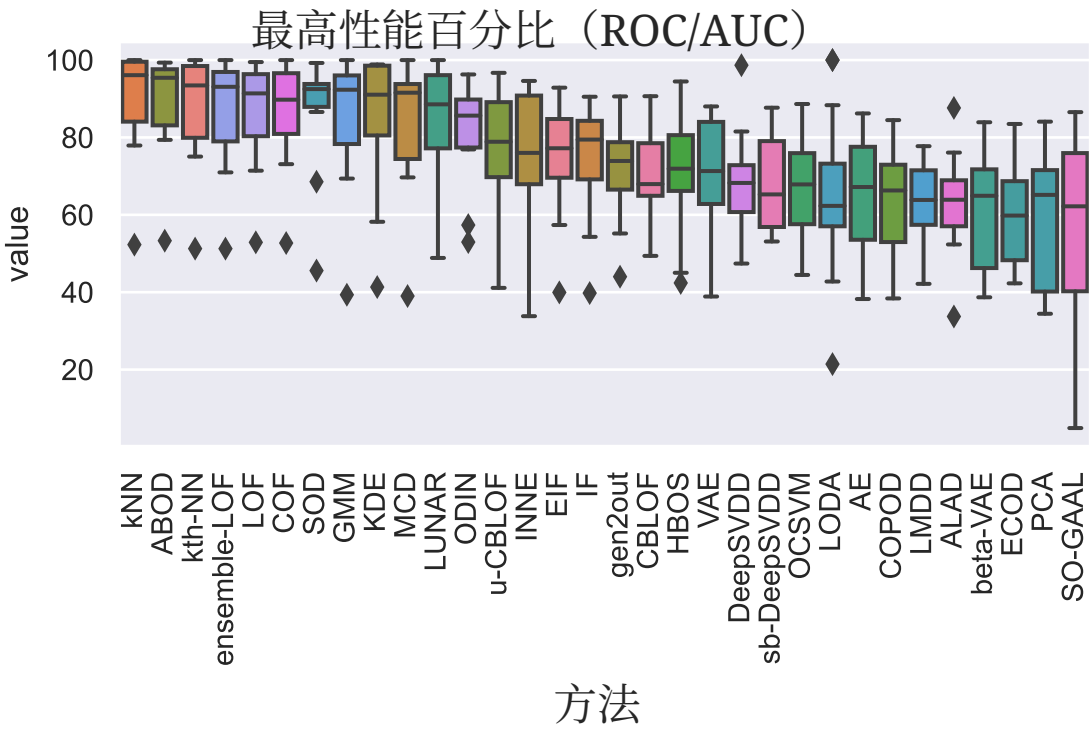


图4：各算法在“局部”数据集上的性能箱线图，以最大 AUC 百分比表示。最大 AUC 是在特定数据集上表现最佳的算法所取得的最高 AUC 值。箱线图的须线从上下四分位数向外延伸，长度为四分位距的 1.5 倍。落在须线范围之外的数据集与算法组合以菱形标记。

表 4：基于 Nemenyi事后分析的局部问题集合中算法之间的显著差异。 ++/+ 表示在 $p = 0.05/p = 0.10$下，行算法的表现优于列算法。行和列分别按均值性能降序和升序排列。当列算法在 $p = 0.05$ 或 $p = 0.10$下不被任何其他算法超越时，不显示该列。最后一列显示均值 AUC。

	SO-GAAL	PCA	ECOD	beta-VAE	ALAD	LMDD	COPOD	AE	LODA	OCSVM	sb-DeepSVDD	DeepSVDD	VAE	HBOS	CBLOF	gen2out	**Mean AUC**
kNN	++	++	++	++	++	++	++	++	++	++	++	++	++	++	++	++	0.740
ABOD	++	++	++	++	++	++	++	++	++	++	++	++	++	++	++	++	0.733
kth-NN	++	++	++	++	++	++	++	++	++	++	++	+					0.722
集成LOF	++	++	++	++	++	++	++	++									0.714
LOF	++	++	++	++	++	++	++	++	+	+	+						0.713
COF	++	++	++	++	++	++	++	++									0.713
SOD	++	++	++	++	++	++	++	++		+	+						0.711
GMM	++	++	++	++		++											0.701
KDE	++	++	++	++	++	++	++	++			+						0.693
MCD		++	+	+													0.687
LUNAR	++	++	++	++		++											0.677
ODIN		+															0.672
u-CBLOF																	0.635
INNE																	0.624
EIF																	0.614
IF																	0.607
gen2out																	0.585
CBLOF																	0.577
HBOS																	0.575
VAE																	0.563
DeepSVDD																	0.557
sb-DeepSVDD																	0.546
单类支持向量机																	0.543
LODA																	0.534
AE																	0.528
COPOD																	0.518
LMDD																	0.516
ALAD																	0.511
β变分自编码器																	0.499
ECOD																	0.492
PCA																	0.475
SO-GAAL																	0.423

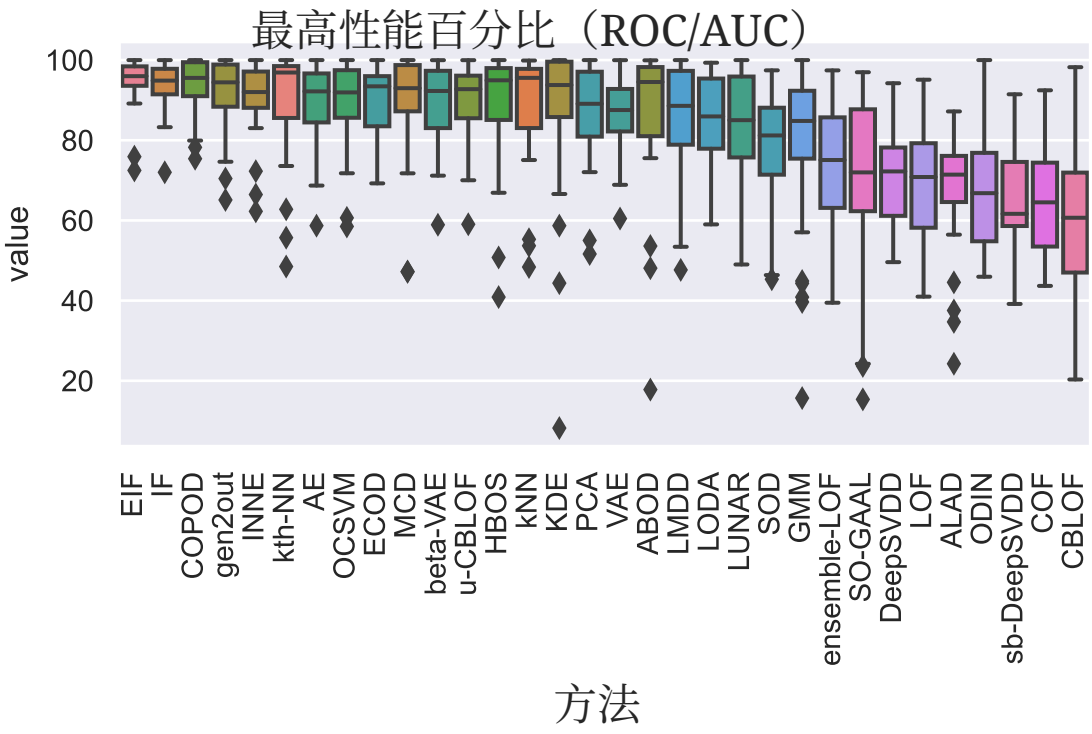


图5：各算法在全局数据集上的性能箱线图，以最大 AUC 百分比表示。最大 AUC 是在特定数据集上表现最佳的算法所取得的最高 AUC 值。箱线图的须线延伸至低四分位数和高四分位数之外，长度为四分位距的 1.5 倍。位于须线范围之外的数据集与算法组合以菱形标记。

表5：基于Nemenyi事后分析，对全局问题集合中算法之间的显著差异进行比较。 ++/+ 表示行算法在 $p = 0.05/p = 0.10$下优于列算法。行和列分别按均值性能降序和升序排列。当没有其他算法在 $p = 0.05$ 或 $p = 0.10$下优于列算法时，不显示该列。最后一列显示平均AUC。

	CBLOF	COF	sb-DeepSVDD	ODIN	ALAD	LOF	DeepSVDD	SO-GAAL	ensemble-LOF	GMM	SOD	LUNAR	LODA	LMDD	PCA	**Mean AUC**
EIF	++	++	++	++	++	++	++	++	++	++	++	++	+	+	0.824	
IF	++	++	++	++	++	++	++	++	++	+	++	++	+			0.814
COPOD	++	++	++	++	++	++	++	++	++	+	++	++				0.809
gen2out	++	++	++	++	++	++	++	++	++		++					0.801
INNE	++	++	++	++	++	++	++	++	++		++					0.790
kth-NN	++	++	++	++	++	++	++	++	++		++					0.785
AE	++	++	++	++	++	++	++	++	++							0.782
单类支持向量机	++	++	++	++	++	++	++	++	++		++					0.781
ECOD	++	++	++	++	++	++	++	++	++		++					0.780
MCD	++	++	++	++	++	++	++	++	++		++					0.779
β变分自编码器	++	++	++	++	++	++	++	++	+							0.779
u-CBLOF	++	++	++	++	++	++	++	++	+							0.776
HBOS	++	++	++	++	++	++	++	++	++		++					0.774
kNN	++	++	++	++	++	++	++	++	++		++					0.774
KDE	++	++	++	++	++	++	++	++	++		++					0.766
PCA	++	++	++	++	++	++	++									0.763
VAE	++	++	++	++	++	++	++	+								0.758
ABOD	++	++	++	++	++	++	++	++								0.752
LMDD	++	++	++	++	++	++	++									0.738
LODA	++	++	++	++			+									0.734
LUNAR	++	++	++	++												0.722
SOD																0.679
GMM	++	++	++	++												0.677
集成LOF																0.643
SO-GAAL																0.609
DeepSVDD													-	--	--	0.605
LOF														--	--	0.603
ALAD														--	--	0.588
ODIN										--		--	--	--	--	0.571
sb-DeepSVDD										--		--	--	--	--	0.556
COF										--		--	--	--	--	0.553
CBLOF										--		--	--	--	--	0.500

5. 讨论

在我们的研究中，我们比较了异常检测算法在 52 个具有语义意义的真实世界表格数据集上的性能，数量超过了其他近期比较研究（Campos 等人（2016）；Goldstein 和 Uchida，2016；Xu 等，2018；Soenen 等，2021；Steinbuss 和 B¨ohm，2021；Domingues 等，2018；Han 等，2022）。Fern´andez-Delgado 等人（2014）对分类算法开展的一项较为类似的研究，轻松地纳入了 121 个数据集。这种差异的主要原因是，用于比较异常检测算法的数据集很少包含分类变量，而分类变量对于比较分类算法并不是问题。当然，也可以进一步扩展数据集集合，例如通过修改数据集。Campos 等（2016）、Emmott 等（2013、2015）以及 Steinbuss 和 B¨ohm（2021）以不同方式修改数据集，从单个基础数据集中创建了具有不同特征的相似数据集。尽管此类修改对于有针对性的研究可能很有用，但近乎重复的数据集彼此之间远非独立，因此似乎不利于对异常检测算法进行适当的统计比较，Emmott 等（2015）的研究就体现了这一点。

在本研究中，我们比较了 32 种最常用的异常检测算法。这一集合当然并不全面：还存在许多其他方法（Schubert 和 Zimek，2019年；Goldstein 和 Uchida，2016年；Emmott 等，2015年；Ruff 等，2021年；Domingues 等，2018年），而且很可能还会有更多方法被发明出来。在这一方面，与 Fern´ández-Delgado 等人（2014年）关于分类算法的研究相比，也存在明显差异；该研究纳入了来自 17 个不同家族的 179 个分类器。显然，分类算法的数量大大超过了异常检测算法的数量。但也许更重要的是，在许多不同的机器学习库中，分类算法有更多成熟且易于使用的实现，而异常检测算法的开箱即用实现则少得多。为了能够开展本研究中的比较，我们不得不花费相当多的精力来清理现有代码，并有时重新实现其中的部分代码。

在本研究中，我们选择不涉及集成的元技术。尽管集成方法非常值得关注，但在着手研究集成方法之前，深入了解基学习器的性能是必不可少的前提。

虽然我们在比较中评估了神经网络，但目前尚无针对任意给定数据集构建高性能网络的通用指南，而这对于本研究所考虑的无监督设置至关重要。此外，许多此类方法的优势在于网络隐式执行的高层次特征提取，而这一优势无法在本基准测试所使用的规模较小的表格数据集上得到充分发挥。与 Ruff 等人（2021年）一样，我们注意到，适用于深度学习方法的分类基准数据集与异常检测基准数据集之间存在显著差距。如果能为该领域增加更多适用于基于深度学习的异常检测的基准数据集，将会是一个值得欢迎的进展。

对数据集上的算法性能进行交叉比较后，我们发现两簇数据集之间存在明显的区分，对应的算法也大致分为两簇。参照文献中的常用术语（Breunig 等人，2000年；Goldstein 和 Uchida，2016年），我们将这些簇分别称为“局部”和“全局”数据集与算法。不过，我们很清楚，在更细致地分析更多数据集和算法时，这种划分可能会显得过于简单。

例如，局部问题和全局问题可能存在相当大的重叠，但未必分别与多模态和单模态异常检测问题完全等同。当不同模态的密度开始出现差异时，多模态问题与局部问题便会有所重叠；此时，试图估计局部密度的局部算法会优于无法区分这些差异的全局算法。还需要开展更多理论和实证研究，例如在精心设计的模拟数据集上进行研究，以进一步阐明这一问题。

6. 结论

根据我们的研究，我们可以制定一般性指南，说明用户应在何种情况下为其问题应用哪些异常检测方法。

一般来说，如果用户事先不了解其数据集是否包含局部异常或全局异常，那么 k-thNN 是最佳选择。在 $p = 0.05$ 方面，它的表现优于评估的 32 种其他方法中的 14 种，并且根据 AUC 均值分数，它也是表现最佳的方法之一。如果 kNN 的 $\mathcal{O}(n^2)$（或优化后的 $\mathcal{O}(n \log\ n)$）计算复杂度是个问题，那么表现稍逊的孤立森林或其扩展变体都是不错的替代方案。

当已知某个数据集包含局部异常时，表现最佳的方法是kNN，它在 $p = 0.10$ 方面优于 32 种方法中的 16 种。

对于仅包含全局异常的数据集，最好使用 IF（孤立森林）进行分析，或者更理想地使用其扩展变体 EIF；后者是在包含全局异常的数据集上表现最佳的算法。COPOD、gen2out、INNE 和 k-thNN 的表现相当，而且这些方法在 $p = 0.05$ 方面均优于至少 10 种其他方法。IF 和 EIF 是计算复杂度最低的算法，在实践中通常更值得优先采用。

综合考虑以上因素，我们倾向于用“三种”来回答论文标题中的问题：包含 k-thNN、 kNN 和 EIF 的工具箱，似乎足以在本研究所考察的多变量数据集上取得良好表现。鉴于本研究的范围，这三种算法很可能在未见过的真实世界多变量数据集上也表现良好。若将其他算法和/或数据集纳入其中，这一结论仍有待进一步考察；扩展我们已开源的代码和数据集预处理流程后，验证起来应该相对容易。后续研究可尝试将本研究的比较分析扩展到原始图像、文本和时间序列等不同类型的数据。

致谢与经费披露

本文所报告的研究部分由荷兰科学研究组织资助项目 NWA.1160.18.238（PrimaVera）以及 BMK、BMDW 和上奥地利州资助，资助项目属于 SCCH 能力中心 INTEGRATE [（FFG 资助项目编号：892418）] ，该中心是 FFG COMET 卓越技术能力中心计划的一部分。

A. 各算法-数据集组合的 AUC 分数

见表6。

B. Nemenyi事后分析结果

表7展示了所有数据集的结果，表8展示了局部数据集的结果，表9展示了全局数据集的结果。

参考文献

D. Agarwal. 交叉分类数据流中的异常检测：一种贝叶斯方法。知识与信息系统，11(1):29–44，2007年。M. Amer 和 M. Goldstein. 基于最近邻和聚类的 Rapidminer 异常检测算法。载于第三届 *RapidMiner* 社区会议（*RCOMM 2012*）论文集，第 1–12 页，2012年。J. An 和 S. Cho. 使用重构概率的基于变分自编码器的异常检测。*IE* 专题讲座，2(1):1–18，2015。A. Arning、R. Agrawal 和 P. Raghavan. 一种用于大型数据库偏差检测的线性方法。*KDD*，1141(50):972–981，1996年。T. R. Bandaragoda、K. M. Ting、D. Albrecht、F. T. Liu、Y. Zhu 和 J. R. Wells. 使用最近邻集成的基于孤立的异常检测。*Computational Intelligence*，34(4):968–998，2018年。Z. Bar-Joseph、D. K. Gifford 和 T. S. Jaakkola. 层次聚类的快速最优叶节点排序。*Bioinformatics*，17(suppl 1):S22–S29，2001年。L. Birg´e 和 Y. Rozenholc. 常规直方图应包含多少个箱？*ESAIM:Probability and Statistics*，10:24–45，2006年。A. Brandsæter、E. Vanem 和 I. K. Glad. 船舶系统运行中的高效在线异常检测。*Expert Systems with Applications*，121:418–437，2019年。M. M. Breunig、H.-P. Kriegel、R. T. Ng 和 J. Sander. LOF：识别基于密度的局部离群点。载于*2000*年 *ACM SIGMOD*数据管理国际会议论文集，第 93–104 页，2000年。G. O. Campos、A. Zimek、J. Sander、R. J. Campello、B. Micenkov´a、E. Schubert、I. Assent 和 M. E. Houle. 无监督离群点检测的评估：度量指标、数据集与实证研究。*Data Mining and Knowledge Discovery*，30(4):891–927，2016年。J. Dem˘sar. 多个数据集上分类器的统计比较。*The Journal ofMachine Learning Research*，7:1–30，2006年。R. Domingues、M. Filippone、P. Michiardi 和 J. Zouaoui. 离群点检测算法的比较评估：实验与分析。*Pattern Recognition*，74:406–421，2018年。

表6：各算法-数据集组合的 AUC 值。



表 7：基于全部 52 个数据集，对所有算法对进行 Nemenyi事后分析得到的 p 值。低于 0.05 的 p 值已用粗体标出。

	高斯混合模型	CBLOF	SO-GAAL	INNE	EIF		sb-DeepSVDD	ALAD	自编码器		MCD	kth-NN	COPOD	LOF		ODIN	集成LOF	多变分自编码器	COF		DeepSVDD	u-CBLOF	单类支持向量机	ABOD	主成分分析		kNN	VAE	LODA	IF		ECOD	LMDD	gen2out	LUNAR	核密度估计		HBOS	SOD	
GMM	1.0	**0.001**	**0.016**	0.9	0.272	**0.001**	**0.002**	0.9	0.9	0.138	0.9	0.878	0.072	0.9	0.9	0.147	**0.007**	0.9	0.9	0.9	0.9	0.272	0.9	0.9	0.827	0.9	0.9	0.9	0.9	0.666	0.9	0.9
CBLOF	**0.001**	1.0	0.9	**0.001**	**0.001**	0.9	0.9	**0.001**	**0.001**	**0.001**	**0.001**	0.725	0.9	0.125	**0.002**	0.9	0.9	**0.001**	**0.001**	**0.001**	0.115	**0.001**	**0.001**	0.088	**0.001**	**0.001**	**0.048**	**0.001**	**0.001**	**0.001**	**0.045**
SO-GAAL	**0.016**	0.9	1.0	**0.001**	**0.001**	0.9	0.9	**0.021**	**0.001**	**0.001**	**0.001**	0.9	0.9	0.64	0.058	0.9	0.9	**0.001**	**0.001**	**0.001**	0.622	**0.001**	**0.034**	0.563	**0.001**	**0.019**	0.433	**0.001**	**0.021**	**0.001**	**0.001**	0.416
INNE	0.9	**0.001**	**0.001**	1.0	0.9	**0.001**	**0.001**	0.9	0.9	0.9	0.9	**0.021**	**0.001**	0.35	0.9	0.9	**0.001**	**0.001**	0.9	0.9	0.9	0.372	0.9	0.9	0.442	0.9	0.9	0.571	0.9	0.9	0.9	0.585
EIF	0.272	**0.001**	**0.001**	0.9	1.0	**0.001**	**0.001**	0.226	0.9	0.9	0.9	**0.001**	**0.001**	**0.002**	0.104	**0.001**	**0.001**	0.9	0.754	0.9	**0.002**	0.9	0.161	**0.003**	0.9	0.241	**0.007**	0.9	0.226	0.9	0.857	**0.007**
sb-DeepSVDD	**0.001**	0.9	0.9	**0.001**	**0.001**	1.0	0.9	**0.001**	**0.001**	**0.001**	**0.001**	0.265	0.9	**0.013**	**0.001**	0.9	0.9	**0.001**	**0.001**	**0.001**	**0.012**	**0.001**	**0.001**	**0.008**	**0.001**	**0.001**	**0.004**	**0.001**	**0.001**	**0.001**	**0.001**	**0.003**
ALAD	**0.002**	0.9	0.9	**0.001**	**0.001**	0.9	1.0	**0.002**	**0.001**	**0.001**	**0.001**	0.9	0.9	0.268	**0.008**	0.9	0.9	**0.001**	**0.001**	**0.001**	0.251	**0.001**	**0.004**	0.203	**0.001**	**0.002**	0.123	**0.001**	**0.002**	**0.001**	**0.001**	0.115
AE	0.9	**0.001**	**0.021**	0.9	0.226	**0.001**	**0.002**	1.0	0.9	0.112	0.9	0.9	0.091	0.9	0.9	0.181	**0.01**	0.9	0.9	0.9	0.9	0.226	0.9	0.9	0.776	0.9	0.9	0.9	0.9	0.615	0.9	0.9
MCD	0.9	**0.001**	**0.001**	0.9	0.9	**0.001**	**0.001**	0.9	1.0	0.9	0.9	**0.012**	**0.001**	0.254	0.9	0.9	**0.001**	**0.001**	0.9	0.9	0.9	0.272	0.9	0.9	0.329	0.9	0.9	0.474	0.9	0.9	0.9	0.49
kth-NN	0.138	**0.001**	**0.001**	0.9	0.9	**0.001**	**0.001**	0.9	0.112	0.9	1.0	0.9	**0.001**	**0.001**	**0.001**	**0.045**	**0.001**	**0.001**	0.776	0.578	0.9	**0.001**	0.9	0.075	**0.001**	0.9	0.121	**0.002**	0.9	0.112	0.9	0.681	**0.002**
COPOD	0.9	**0.001**	**0.001**	0.9	0.9	**0.001**	**0.001**	0.9	0.9	0.9	1.0	**0.042**	**0.001**	0.494	0.9	**0.001**	**0.001**	0.9	0.9	0.9	0.512	0.9	0.9	0.571	0.9	0.9	0.695	0.9	0.9	0.9	0.9	0.71
LOF	0.878	0.725	0.9	**0.021**	**0.001**	0.265	0.9	0.9	**0.012**	**0.001**	**0.042**	1.0	0.9	0.9	0.9	0.9	0.9	0.208	0.407	**0.003**	0.9	**0.001**	0.9	0.9	**0.001**	0.9	0.9	**0.017**	0.9	**0.001**	0.293	0.9
ODIN	0.072	0.9	0.9	**0.001**	**0.001**	0.9	0.9	0.091	**0.001**	**0.001**	**0.001**	0.9	1.0	0.9	0.203	0.9	0.9	**0.002**	**0.006**	**0.001**	0.9	**0.001**	0.133	0.849	**0.001**	0.085	0.725	**0.001**	0.091	**0.001**	**0.003**	0.71
集成LOF	0.9	0.125	0.64	0.35	**0.002**	**0.013**	0.268	0.9	0.254	**0.001**	0.494	0.9	0.9	1.0	0.9	0.9	0.501	0.853	0.9	0.096	0.9	**0.002**	0.9	0.9	**0.044**	0.9	0.9	0.304	0.9	**0.018**	0.9	0.9
多变分自编码器	0.9	**0.002**	0.658	0.9	0.104	**0.001**	**0.008**	0.9	0.9	**0.045**	0.9	0.9	0.203	0.9	1.0	0.354	**0.028**	0.9	0.9	0.747	0.9	0.104	0.9	0.9	0.585	0.9	0.9	0.9	0.9	0.416	0.9	0.9
COF	0.147	0.9	0.9	**0.001**	**0.001**	0.9	0.9	0.181	**0.001**	**0.001**	**0.001**	0.9	0.9	0.9	0.354	1.0	0.9	**0.004**	**0.015**	**0.001**	0.9	**0.001**	0.251	0.9	**0.001**	0.169	0.886	**0.001**	0.181	**0.001**	**0.008**	0.871
DeepSVDD	**0.007**	0.9	0.9	**0.001**	**0.001**	0.9	0.9	**0.01**	**0.001**	**0.001**	**0.001**	0.9	0.9	0.501	**0.028**	0.9	1.0	**0.001**	**0.001**	**0.001**	0.482	**0.001**	**0.016**	0.416	**0.001**	**0.009**	0.279	**0.001**	**0.01**	**0.001**	**0.001**	0.265
u-CBLOF	0.9	**0.001**	**0.001**	0.9	0.9	**0.001**	**0.001**	0.9	0.9	0.776	0.9	0.208	**0.002**	0.853	0.9	**0.004**	**0.001**	1.0	0.9	0.9	0.871	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9
单类支持向量机	0.9	**0.001**	**0.001**	0.9	0.754	**0.001**	**0.001**	0.9	0.9	0.578	0.9	0.407	**0.006**	0.9	0.9	**0.015**	**0.001**	0.9	1.0	0.9	0.9	0.754	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9
ABOD	0.9	**0.001**	**0.001**	0.9	0.9	**0.001**	**0.001**	0.9	0.9	0.9	0.9	**0.003**	**0.001**	0.096	0.747	**0.001**	**0.001**	0.9	0.9	1.0	0.104	0.9	0.849	0.133	0.9	0.9	0.22	0.9	0.9	0.9	0.9	0.232
PCA	0.9	0.115	0.622	0.372	**0.002**	**0.012**	0.251	0.9	0.272	**0.001**	0.512	0.9	0.9	0.9	0.9	0.9	0.482	0.871	0.9	0.104	1.0	**0.002**	0.9	0.9	**0.048**	0.9	0.9	0.322	0.9	**0.02**	0.9	0.9
kNN	0.272	**0.001**	**0.001**	0.9	0.9	**0.001**	**0.001**	0.226	0.9	0.9	0.9	**0.001**	**0.001**	**0.002**	0.104	**0.001**	**0.001**	0.9	0.754	0.9	**0.002**	1.0	0.161	**0.003**	0.9	0.241	**0.007**	0.9	0.226	0.9	0.857	**0.007**
VAE	0.9	**0.001**	**0.034**	0.9	0.161	**0.001**	**0.004**	0.9	0.9	0.075	0.9	0.9	0.133	0.9	0.9	0.251	**0.016**	0.9	0.9	0.849	0.9	0.161	1.0	0.9	0.688	0.9	0.9	0.9	0.9	0.527	0.9	0.9
LODA	0.9	0.088	0.563	0.442	**0.003**	**0.008**	0.203	0.9	0.329	**0.001**	0.571	0.9	0.849	0.9	0.9	0.9	0.9	0.416	0.9	0.9	0.133	0.9	**0.003**	0.9	1.0	0.065	0.9	0.9	0.389	0.9	**0.028**	0.9	0.9
IF	0.827	**0.001**	**0.001**	0.9	0.9	**0.001**	**0.001**	0.776	0.9	0.9	0.9	**0.001**	**0.001**	**0.044**	0.585	**0.001**	**0.001**	0.9	0.9	0.9	**0.048**	0.9	0.688	0.065	1.0	0.794	0.115	0.9	0.776	0.9	0.9	0.123
ECOD	0.9	**0.001**	**0.019**	0.9	0.241	**0.001**	**0.002**	0.9	0.9	0.121	0.9	0.9	0.085	0.9	0.9	0.169	**0.009**	0.9	0.9	0.9	0.9	0.241	0.9	0.9	0.794	1.0	0.9	0.9	0.9	0.633	0.9	0.9
LMDD	0.9	**0.048**	0.433	0.571	**0.007**	**0.004**	0.123	0.9	0.474	**0.002**	0.695	0.9	0.725	0.9	0.9	0.886	0.279	0.9	0.9	0.9	0.22	0.9	**0.007**	0.9	0.9	0.115	0.9	1.0	0.527	0.9	0.054	0.9	0.9
gen2out	0.9	**0.001**	**0.001**	0.9	0.9	**0.001**	**0.001**	0.9	0.9	0.9	0.9	**0.017**	**0.001**	0.304	0.9	**0.001**	**0.001**	0.9	0.9	0.9	0.322	0.9	0.9	0.389	0.9	0.9	0.527	1.0	0.9	0.9	0.9	0.541
LUNAR	0.9	**0.001**	**0.021**	0.9	0.226	**0.001**	**0.002**	0.9	0.9	0.112	0.9	0.9	0.091	0.9	0.9	0.181	**0.01**	0.9	0.9	0.9	0.9	0.226	0.9	0.9	0.776	0.9	0.9	0.9	1.0	0.615	0.9	0.9
KDE	0.666	**0.001**	**0.001**	0.9	0.9	**0.001**	**0.001**	0.615	0.9	0.9	0.9	**0.001**	**0.001**	**0.018**	0.416	**0.001**	**0.001**	0.9	0.9	0.9	**0.02**	0.9	0.527	**0.028**	0.9	0.633	0.054	0.9	0.615	1.0	0.9	0.058
HBOS	0.9	**0.001**	**0.001**	0.9	0.857	**0.001**	**0.001**	0.9	0.9	0.681	0.9	0.293	**0.003**	0.9	0.9	**0.008**	**0.001**	0.9	0.9	0.9	0.9	0.857	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	1.0	0.9
SOD	0.9	**0.045**	0.416	0.585	**0.007**	**0.003**	0.115	0.9	0.49	**0.002**	0.71	0.9	0.71	0.9	0.9	0.871	0.265	0.9	0.9	0.9	0.232	0.9	**0.007**	0.9	0.9	0.123	0.9	0.9	0.541	0.9	0.058	0.9	1.0

表 8：基于 12 个局部数据集，对所有算法对进行 Nemenyi事后分析所得的 p 值。低于 0.05 的 p 值以粗体显示。

	高斯混合模型	CBLOF	SO-GAAL	INNE	EIF		sb-DeepSVDD	ALAD	自编码器		MCD	kth-NN	COPOD	LOF		ODIN	集成LOF	多变分自编码器	COF		DeepSVDD	u-CBLOF	单类支持向量机	ABOD	主成分分析		kNN	VAE	LODA	核密度估计		ECOD	LMDD	gen2out	LUNAR	核密度估计		HBOS	SOD	
GMM	1.0	0.855	**0.039**	0.9	0.9	0.434	0.128	0.155	0.9	0.9	0.125	0.9	0.9	0.9	**0.034**	0.9	0.647	0.9	0.452	0.9	**0.01**	0.9	0.81	0.499	0.9	0.9	**0.026**	**0.039**	0.869	0.9	0.9	0.855	0.9	
CBLOF	0.855	1.0	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.165	0.9	0.364	0.9	0.476	0.9	0.559	0.9	0.9	0.9	**0.029**	0.9	**0.021**	0.9	0.9	0.9	0.9	0.9	0.9	0.855	0.452	0.9	0.434	
SO-GAAL	**0.039**	0.9	1.0	0.795	0.9	0.9	0.9	0.9	0.107	**0.001**	0.9	**0.002**	0.232	**0.004**	0.9	**0.007**	0.9	0.633	0.9	**0.001**	0.9	**0.001**	0.9	0.9	0.9	0.9	0.9	**0.039**	**0.004**	0.9	**0.003**		
INNE	0.9	0.9	0.795	1.0	0.9	0.9	0.9	0.9	0.9	0.825	0.9	0.9	0.9	0.9	0.766	0.9	0.9	0.9	0.9	0.452	0.559	0.381	0.9	0.9	0.721	0.795	0.9	0.9	0.9	0.9	0.9	0.9	
EIF	0.9	0.9	0.9	0.9	1.0	0.9	0.9	0.9	0.9	0.692	0.9	0.9	0.9	0.9	0.899	0.9	0.9	0.9	0.9	0.9	0.3	0.692	0.244	0.9	0.9	0.9	0.855	0.9	0.9	0.9	0.9	0.9	
sb-DeepSVDD	0.434	0.9	0.9	0.9	0.9	1.0	0.9	0.9	0.647	**0.022**	0.9	0.071	0.84	0.11	0.9	0.155	0.9	0.9	0.9	**0.002**	0.9	**0.002**	0.9	0.9	0.9	0.9	0.9	0.9	0.434	0.099	0.9	0.093	
ALAD	0.128	0.9	0.9	0.9	0.9	0.9	1.0	0.9	0.285	**0.003**	0.9	**0.011**	0.499	**0.02**	0.9	**0.031**	0.9	0.884	0.9	**0.001**	0.9	**0.001**	0.9	0.9	0.9	0.9	0.9	0.9	0.128	**0.017**	0.9	**0.016**	
AE	0.155	0.9	0.9	0.9	0.9	0.9	0.9	1.0	0.33	**0.004**	0.9	**0.015**	0.544	**0.025**	0.9	**0.039**	0.9	0.9	0.9	**0.001**	0.9	**0.001**	0.9	0.9	0.9	0.9	0.9	0.9	0.155	**0.022**	0.9	**0.021**	
MCD	0.9	0.9	0.107	0.9	0.9	0.647	0.285	0.33	1.0	0.9	0.278	0.9	0.9	0.9	0.093	0.9	0.855	0.9	0.662	0.9	**0.034**	0.9	0.9	0.707	0.9	0.076	0.107	0.9	0.9	0.9	0.9	0.9	
kth-NN	0.9	0.165	**0.001**	0.825	0.692	**0.022**	**0.003**	**0.004**	0.9	1.0	**0.003**	0.9	0.9	0.9	**0.001**	0.9	0.066	0.9	**0.024**	0.9	**0.001**	0.9	0.136	**0.031**	0.588	**0.001**	**0.001**	0.175	0.9	0.9	0.165	0.9	
COPOD	0.125	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.278	**0.003**	1.0	**0.011**	0.492	**0.019**	0.9	**0.03**	0.9	0.877	0.9	**0.001**	0.9	**0.001**	0.9	0.9	0.9	0.9	0.9	0.9	0.125	**0.017**	0.9	**0.015**	
LOF	0.9	0.364	**0.002**	0.9	0.9	0.071	**0.011**	**0.015**	0.9	0.9	**0.011**	1.0	0.9	0.9	**0.002**	0.9	0.175	0.9	0.076	0.9	**0.001**	0.9	0.314	0.093	0.81	**0.001**	**0.002**	0.381	0.9	0.9	0.364	0.9	
ODIN	0.9	0.9	0.232	0.9	0.9	0.84	0.499	0.544	0.9	0.9	0.492	0.9	1.0	0.9	0.208	0.9	0.9	0.9	0.855	0.9	0.087	0.9	0.9	0.899	0.9	0.175	0.232	0.9	0.9	0.9	0.9	0.9	
集成LOF	0.9	0.476	**0.004**	0.9	0.9	0.11	**0.02**	**0.025**	0.9	0.9	**0.019**	0.9	0.9	1.0	**0.004**	0.9	0.251	0.9	0.118	0.9	**0.001**	0.9	0.426	0.141	0.9	**0.003**	**0.004**	0.492	0.9	0.9	0.476	0.9	
多变分自编码器	**0.034**	0.9	0.9	0.766	0.899	0.9	0.9	0.9	0.093	**0.001**	0.9	**0.002**	0.208	**0.004**	1.0	**0.006**	0.9	0.603	0.9	**0.001**	0.9	**0.001**	0.9	0.9	0.9	0.9	0.9	**0.034**	**0.003**	0.9	**0.003**		
COF	0.9	0.559	**0.007**	0.9	0.9	0.155	**0.031**	**0.039**	0.9	0.9	**0.03**	0.9	0.9	0.9	**0.006**	1.0	0.33	0.9	0.165	0.9	**0.002**	0.9	0.514	0.196	0.9	**0.005**	**0.007**	0.573	0.9	0.9	0.559	0.9	
DeepSVDD	0.647	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.855	0.066	0.9	0.175	0.9	0.251	0.9	0.33	1.0	0.9	0.9	**0.009**	0.9	**0.006**	0.9	0.9	0.9	0.9	0.9	0.647	0.232	0.9	0.219		
u-CBLOF	0.9	0.9	0.633	0.9	0.9	0.9	0.884	0.9	0.9	0.877	0.9	0.9	0.9	0.9	0.603	0.9	0.9	1.0	0.9	0.618	0.381	0.559	0.9	0.9	0.9	0.559	0.633	0.9	0.9	0.9	0.9	0.9	
单类支持向量机	0.452	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.662	**0.024**	0.9	0.076	0.855	0.118	0.9	0.165	0.9	0.9	1.0	**0.003**	0.9	**0.002**	0.9	0.9	0.9	0.9	0.9	0.9	0.452	0.107	0.9	0.099	
ABOD	0.9	**0.029**	**0.001**	0.452	0.3	**0.002**	**0.001**	**0.001**	0.9	0.9	**0.001**	0.9	0.9	0.9	**0.001**	0.9	**0.009**	0.618	**0.003**	1.0	**0.001**	0.9	**0.022**	**0.003**	0.208	**0.001**	**0.001**	**0.031**	0.9	0.9	**0.029**	0.9	
PCA	**0.01**	0.9	0.9	0.559	0.692	0.9	0.9	0.9	**0.034**	**0.001**	0.9	**0.003**	0.087	**0.001**	0.9	**0.002**	0.9	0.381	0.9	**0.001**	1.0	**0.001**	0.9	0.9	0.795	0.9	0.9	0.9	**0.01**	**0.001**	0.9	**0.001**	
kNN	0.9	**0.021**	**0.001**	0.381	0.244	**0.002**	**0.001**	**0.001**	0.9	0.9	**0.001**	0.9	0.9	0.9	**0.001**	0.9	**0.006**	0.559	**0.002**	0.9	**0.001**	1.0	**0.016**	**0.002**	0.165	**0.001**	**0.001**	**0.022**	0.9	0.9	**0.021**	0.9	
VAE	0.81	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.136	0.9	0.314	0.9	0.426	0.9	0.514	0.9	0.9	0.9	0.9	**0.022**	0.9	**0.016**	1.0	0.9	0.9	0.9	0.9	0.81	0.399	0.9	0.381		
LODA	0.499	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.707	**0.031**	0.9	0.093	0.839	0.141	0.9	0.196	0.9	0.9	0.9	**0.003**	0.9	**0.002**	0.9	1.0	0.9	0.9	0.9	0.499	0.128	0.9	0.122		
IF	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.588	0.9	0.81	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.208	0.795	0.165	0.9	0.9	1.0	0.9	0.9	0.9	0.884	0.9	0.869		
ECOD	**0.026**	0.9	0.9	0.721	0.855	0.9	0.9	0.9	0.076	**0.001**	0.9	**0.001**	0.175	**0.003**	0.9	**0.005**	0.9	0.559	0.9	**0.001**	0.9	**0.001**	0.9	0.9	0.9	1.0	0.9	0.9	**0.026**	**0.002**	0.9	**0.002**	
LMDD	**0.039**	0.9	0.9	0.795	0.9	0.9	0.9	0.9	0.107	**0.001**	0.9	**0.002**	0.232	**0.004**	0.9	**0.007**	0.9	0.633	0.9	**0.001**	0.9	**0.001**	0.9	0.9	0.9	0.9	1.0	0.9	**0.039**	**0.004**	0.9	**0.003**	
gen2out	0.869	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.175	0.9	0.381	0.9	0.492	0.9	0.573	0.9	0.9	0.9	**0.031**	0.9	**0.022**	0.9	0.9	0.9	0.9	0.9	1.0	0.869	0.468	0.9	0.452	
LUNAR	0.9	0.855	**0.039**	0.9	0.9	0.434	0.128	0.155	0.9	0.9	0.125	0.9	0.9	0.9	**0.034**	0.9	0.647	0.9	0.452	0.9	**0.01**	0.9	0.81	0.499	0.9	**0.026**	**0.039**	0.869	1.0	0.9	0.855	0.9	
KDE	0.9	0.452	**0.004**	0.9	0.9	0.099	**0.017**	**0.022**	0.9	0.9	**0.017**	0.9	0.9	0.9	**0.003**	0.9	0.232	0.9	0.107	0.9	**0.001**	0.9	0.399	0.128	0.884	**0.002**	**0.004**	0.468	0.9	1.0	0.452	0.9	
HBOS	0.855	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.165	0.9	0.364	0.9	0.476	0.9	0.559	0.9	0.9	0.9	**0.029**	0.9	**0.021**	0.9	0.9	0.9	0.9	0.9	0.9	0.855	0.452	1.0	0.434	
SOD	0.9	0.434	**0.003**	0.9	0.9	0.093	**0.016**	**0.021**	0.9	0.9	**0.015**	0.9	0.9	0.9	**0.003**	0.9	0.219	0.9	0.099	0.9	**0.001**	0.9	0.381	0.122	0.869	**0.002**	**0.003**	0.452	0.9	0.9	0.434	1.0	

表 9：基于 37 个全局数据集，对所有算法对进行 Nemenyi事后分析 得到的 p 值。小于 0.05 的 p 值以粗体显示。

	GMM	CBLOF	SO-GAAL	INNE	EIF		sb-DeepSVDD	ALAD	自编码器		MCD	kth-NN	COPOD	LOF		ODIN	集成LOF	多变分自编码器	COF		DeepSVDD	u-CBLOF	单类支持向量机	ABOD	主成分分析		kNN	VAE	LODA	IF		ECOD	LMDD	gen2out	LUNAR	核密度估计		HBOS	SOD	
GMM	1.0	**0.007**	0.757	0.613	**0.002**	**0.001**	0.2	0.9	0.9	0.165	0.063	0.35	**0.034**	0.9	0.9	0.791	0.9	0.9	0.791	0.9	0.9	0.605	0.9	0.9	0.052	0.9	0.9	0.187	0.9	0.622	0.9	0.9	
CBLOF	**0.007**	1.0	0.9	**0.001**	**0.001**	0.9	0.9	**0.001**	**0.001**	**0.001**	**0.001**	0.9	0.866		**0.001**	0.9	0.9	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.004**	**0.001**	**0.001**	**0.001**	**0.001**	**0.01**	**0.001**	**0.001**	0.639		
SO-GAAL	0.757	0.9	1.0	**0.001**	**0.001**	0.9	0.9	**0.005**	**0.001**	**0.001**	**0.001**	0.9	0.9	0.9	**0.006**	0.9	0.9	**0.007**	**0.001**	**0.014**	0.2	**0.001**	0.073	0.647	**0.001**	**0.001**	0.148	**0.001**	0.816	**0.001**	**0.001**	0.9	
INNE	0.613	**0.001**	**0.001**	1.0	0.9	**0.001**	**0.001**	0.9	0.9	0.9	0.9	**0.001**	**0.001**	**0.001**	0.9	**0.001**	**0.001**	0.9	0.9	0.9	0.9	0.9	0.9	0.723	0.9	0.9	0.9	0.9	0.554	0.9	0.9	**0.006**	
EIF	**0.002**	**0.001**	**0.001**	0.9	1.0	**0.001**	**0.001**	0.613	0.892	0.9	0.9	**0.001**	**0.001**	**0.001**	0.58	**0.001**	**0.001**	0.554	0.9	0.44	0.05	0.9	0.148	**0.004**	0.9	0.845	0.073	0.9	**0.001**	0.9	0.9	**0.001**	
sb-DeepSVDD	**0.001**	0.9	0.9	**0.001**	**0.001**	1.0	0.9	**0.001**	**0.001**	**0.001**	**0.001**	0.9	0.9	0.597	**0.001**	0.9	0.9	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.002**	**0.001**	**0.001**	**0.001**	0.35	
ALAD	0.2	0.9	0.9	**0.001**	**0.001**	0.9	1.0	**0.001**	**0.001**	**0.001**	0.9	0.9	0.9	0.9	**0.001**	0.9	0.9	**0.001**	**0.001**	**0.001**	**0.014**	**0.001**	**0.003**	0.128	**0.001**	**0.001**	**0.009**	**0.001**	0.249	**0.001**	**0.001**	0.9	
AE	0.9	**0.001**	**0.005**	0.9	0.613	**0.001**	**0.001**	1.0	0.9	0.9	0.9	**0.001**	**0.001**	**0.048**	0.9	0.9	**0.001**	**0.001**	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.137	
MCD	0.9	**0.001**	**0.001**	0.9	0.892	**0.001**	**0.001**	0.9	1.0	0.9	0.9	**0.001**	**0.001**	**0.01**	0.9	0.9	**0.001**	**0.001**	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	**0.037**	
kth-NN	0.165	**0.001**	**0.001**	0.9	0.9	**0.001**	**0.001**	0.9	0.9	1.0	0.9	**0.001**	**0.001**	**0.001**	0.9	**0.001**	**0.001**	0.9	0.9	0.9	0.9	0.706	0.9	0.9	0.249	0.9	0.9	0.782	0.9	0.128	0.9	0.9	**0.001**
COPOD	0.063	**0.001**	**0.001**	0.9	0.9	**0.001**	**0.001**	0.9	0.9	0.9	1.0	**0.001**	**0.001**	**0.001**	0.9	**0.001**	**0.001**	0.9	0.9	0.

D. Dua 和 C. Graff。UCI机器学习库，2017年。网址 http://archive.ics. uci.edu/ml。A. Emmott、S. Das、T. Dietterich、A. Fern 和 W.-K. Wong。异常检测问题的元分析。*arXiv* 预印本 *arXiv:1503.01158*，2015年。A. F. Emmott、S. Das、T. Dietterich、A. Fern 和 W.-K. Wong。基于真实数据系统化构建异常检测基准测试。载于*ACM SIGKDD* 离群值检测与描述研讨会论文集，第 16–21 页，2013年。M. Fern´andez-Delgado、E. Cernadas、S. Barro 和 D. Amorim。解决现实世界的分类问题是否需要数百个分类器? *The Journal of Machine LearningResearch*，15(1):3133–3181，2014年。R. Foorthuis。异常的性质与类型：数据偏差综述。*International Journal of Data Science and Analytics*，12(4):297–331，2021年。M. Goldstein 和 A. Dengel。基于直方图的离群值评分（HBOS）：一种快速的无监督异常检测算法。*KI-2012*：海报与演示专题，第 59–63 页，2012年。M. Goldstein 和 S. Uchida。多变量数据无监督异常检测算法的比较评估。*PloS one*，11(4):e0152173，2016年。A. Goodge、B. Hooi、S.-K. Ng 和 W. S. Ng。LUNAR：通过图神经网络统一局部离群值检测方法。载于*AAAI*人工智能会议论文集，第 36 卷，第 6737–6745 页，2022年。S. Han、X. Hu、H. Huang、M. Jiang 和 Y. Zhao。ADBench：异常检测基准测试。载于神经信息处理系统（*NeurIPS*），2022年。S. Hariri、M. C. Kind 和 R. J. Brunner。扩展隔离森林。*IEEE*知识与数据工程汇刊，33(4):1479–1489，2019年。V. Hautamaki、I. Karkkainen 和 P. Franti。使用 k 近邻图进行离群值检测。载于第*17*届国际模式识别会议论文集，*2004*。*ICPR 2004*。，第 3 卷，第 430–433 页。IEEE，2004年。R. He 和 J. McAuley。起伏变化：利用单类协同过滤对时尚趋势的视觉演变进行建模。载于第*25*届国际万维网会议论文集，第 507–517 页，2016年。Z. He、X. Xu 和 S. Deng。发现基于聚类的局部离群值。*Pattern RecognitionLetters*，24(9-10):1641–1650，2003年。R. L. Iman 和 J. M. Davenport。对 fbietkan 统计量临界区域的近似。*Communications in Statistics - Theory and Methods*，9(6):571–595，1980年。N. Japkowicz、C. Myers、M. Gluck 等。用于分类的新颖性检测方法。载于*IJCAI*，第 1 卷，第 518–523 页。Citeseer，1995年。

H.-P. Kriegel、M. Schubert 和 A. Zimek。高维数据中的基于角度的异常值检测。载于第*14*届*ACM SIGKDD*国际知识发现与数据挖掘会议论文集，第444–452页，2008年。H.-P. Kriegel、P. Kr¨oger、E. Schubert 和 A. Zimek。高维数据轴平行子空间中的异常值检测。载于亚太知识发现与数据挖掘会议，第831–838页。Springer，2009年。A. Krizhevsky、G. Hinton 等。利用微型图像学习多层特征。2009年。L. J. Latecki、A. Lazarevic 和 D. Pokrajac。利用核密度函数进行异常值检测。载于模式识别中的机器学习与数据挖掘国际研讨会，第61–75页。Springer，2007年。M.-C. Lee、S. Shekhar、C. Faloutsos、T. N. Hutson 和 L. Iasemidis。Gen 2 out：检测并排序广义异常。载于*2021*年*IEEE*国际大数据会议（*Big Data*），第801–811页。IEEE，2021年。Z. Li、Y. Zhao、N. Botta、C. Ionescu 和 X. Hu。COPOD：基于Copula的异常值检测。载于*2020*年*IEEE*国际数据挖掘会议（*ICDM*），第1118–1123页。IEEE，2020。Z. Li、Y. Zhao、X. Hu、N. Botta、C. Ionescu 和 G. H. Chen。ECOD：利用经验累积分布函数进行无监督异常值检测，2022年。F. T. Liu、K. M. Ting 和 Z.-H. Zhou。孤立森林。载于*2008*年第八届*IEEE*国际数据挖掘会议，第413–422页。IEEE，2008年。Y. Liu、Z. Li、C. Zhou、Y. Jiang、J. Sun、M. Wang 和 X. He。用于无监督异常值检测的生成对抗主动学习。*IEEE*知识与数据工程汇刊，32(8)：1517–1528，2019年。K. Malik、H. Sadawarti 和 K. G S. 异常值检测技术的比较分析。《国际计算机应用杂志》，97(8)：12–21，2014年。P. B. Nemenyi。无分布假设的多重比较。普林斯顿大学，1963年。T. Pevn`y。LODA：轻量级在线异常检测器。《机器学习》，102(2)：275–304，2016年。S. Ramaswamy、R. Rastogi 和 K. Shim。用于从大型数据集中挖掘异常值的高效算法。载于*2000*年*ACM SIGMOD*数据管理国际会议论文集，第427–438页，2000年。S. Rayana。ODDS库，2016年。网址：http://odds.cs.stonybrook.edu。P. J. Rousseeuw 和 C. Croux。中位数绝对偏差的替代方法。《美国统计学会杂志》，88(424)：1273–1283，1993年。P. J. Rousseeuw 和 K. V. Driessen。用于最小协方差行列式估计量的快速算法。《技术计量学》，41(3)：212–223，1999年。

L. Ruff、R. A. Vandermeulen、N. G¨ornitz、A. Binder、E. M¨uller、K.-R. M¨uller 和 M. Kloft。深度半监督异常检测。*arXiv* 预印本 *arXiv:1906.02694*，2019 年。L. Ruff 、J. R. Kauffmann、R. A. Vandermeulen、G. Montavon、W. Samek、M. Kloft、T. G. Dietterich 和 K.-R. M¨uller。深度与浅层异常检测的统一综述。*IEEE*会刊，2021年。B. Sch¨olkopf、R. C. Williamson、A. J. Smola、J. Shawe-Taylor、J. C. Platt 等。用于新颖性检测的支持向量方法。载于*NIPS*，第12卷，第582–588页，1999年。E. Schubert 和 A. Zimek。Elki：用于数据分析的大型开源库。*CoRR*，abs/1902.03616，2019年。网址：https://arxiv.org/abs/1902.03616。S. Y. Shin 和 H.-j. Kim。通过沿投影路径进行重构以实现新颖性检测的扩展自编码器。《应用科学》，10(13):4497，2020。M.-L. Shyu、S.-C. Chen、K. Sarinnapakorn 和 L. Chang。一种基于主成分分类器的新型异常检测方案。技术报告，迈阿密大学电气与计算机工程系，2003年。E. H. Simpson。列联表中交互作用的解释。《皇家统计学会杂志：*B*辑（方法论）》，13(2):238–241，1951年。J. Soenen、E. Van Wolputte、L. Perini、V. Vercruyssen、W. Meert、J. Davis 和 H. Block- eel。超参数调优对无监督异常检测方法比较评估的影响。载于*KDD'21*异常值检测与描述研讨会论文集，第1–9页。异常值检测与描述组织委员会，2021年。G. Steinbuss 和 K. B¨ohm。使用真实感合成数据对无监督离群点检测进行基准测试。《*ACM*数据知识发现汇刊》（*TKDD*），15(4):1–20，2021年。J. Tang、Z. Chen、A. W.-C. Fu 和 D. W. Cheung。提高低密度模式下异常值检测的有效性。载于亚太知识发现与数据挖掘会议，第535–548页。Springer，2002年。A. Thomas、A. Gramfort 和 S. Cl´emen¸con。为无监督异常检测学习超参数。载于*Conf´erence sur L'apprentissage automatique-Cap 2016*，2016年。X. Xu、H. Liu、L. Li 和 M. Yao。高维数据异常值检测技术比较。《国际计算智能系统杂志》，11(1):652–662，2018年。H. Zenati、M. Romain、C.-S. Foo、B. Lecouat 和 V. Chandrasekhar。对抗学习异常检测。载于*2018*年*IEEE*国际数据挖掘会议（*ICDM*），第727–736页。IEEE，2018年。Y. Zhao、Z. Nasrullah 和 Z. Li。PyOD：用于可扩展离群点检测的Python工具箱，2019年。

L. Zhou、W. Deng 和 X. Wu。使用 VAE 和 β变分自编码器进行无监督异常定位。*arXiv* 预印本 *arXiv:2005.10686*，2020。A. Zimek、R. J. Campello 和 J. Sander。无监督离群点检测集成：挑战与研究问题——一篇立场论文。*ACM SIGKDD Explorations Newsletter*，15(1):11–22，2014。