Journal of Image and Signal Processing 图像与信号处理, 2024, 13(1), 92-105
Published Online January 2024 in Hans. https://www.hanspub.org/journal/jisp
https://doi.org/10.12677/jisp.2024.131009

Hans汉斯

基于元学习的自动化异常检测

普文寅[1]，牛少彰[1,2]，安洪旭[2]，史成洁[3]，王茂森[2]

[1]北京邮电大学计算机学院(国家示范性软件学院)，北京
[2]东南数字经济发展研究院，浙江 衢州
[3]中国科学院信息工程研究所，北京

收稿日期：2023年12月16日；录用日期：2024年1月24日；发布日期：2024年1月31日

摘　要

在实际的工业生产环境中，常常需要监控机器相关指标的运行状况，对于一个多变量的无监督异常检测任务，由于缺乏带标签的数据，并且同一个检测算法在不同数据集上的性能表现不同，模型设计依赖于人工调整，所以如何高效选择一个异常检测模型并完成超参数调整成为了一个亟待解决的问题。在这篇文章中，我们建立了一个异常检测模型自动选择机制，称为AutoAD (Auto Anomaly Detector)。AutoAD利用了历史数据上异常检测模型的表现和数据集本身的特征，基于元学习的想法，通过深度神经网络自动选择一个有效的异常检测模型并调优，用于新的数据集的异常检测。实验结果表明了在开源数据集上AutoAD在异常检测模型自动选择方面具有有效性。

关键词

异常检测，自动化机器学习，元学习

Automated Anomaly Detection Based on Meta-Learning

Wenyin Pu[1], Shaozhang Niu[1,2], Hongxu An[2], Chengjie Shi[3], Maosen Wang[2]

[1]School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing
[2]Southeast Digital Economy Development Institute, Quzhou Zhejiang
[3]Institute of Information Engineering, Chinese Academy of Sciences, Beijing

Received: Dec. 16th, 2023; accepted: Jan. 24th, 2024; published: Jan. 31st, 2024

Abstract

In real industrial production environments, it is often necessary to monitor the operating condi-

文章引用: 普文寅, 牛少彰, 安洪旭, 史成洁, 王茂森. 基于元学习的自动化异常检测[J]. 图像与信号处理, 2024, 13(1): 92-105. DOI: 10.12677/jisp.2024.131009

tions of machine-related indicators. For a multivariate unsupervised anomaly detection task, due to the lack of labeled data and the different performance of the same detection algorithm on different datasets, the design of the model relies on manual tuning, so how to efficiently select an anomaly detection model and complete hyper-parameter tuning has become a pressing problem. In this paper, we develop an automatic anomaly detection model selection mechanism called AutoAD (Auto Anomaly Detector). AutoAD utilizes the performance of anomaly detection models on historical data and the characteristics of the dataset itself to automatically select an effective anomaly detection model based on the idea of meta-learning through deep neural networks, and tunes it for anomaly detection on new datasets. The experimental results show that on the open source dataset AutoAD is effective in automatic anomaly detection model selection.

Keywords

Anomaly Detection, Automated Machine Learning, Meta-Learning

1. 前言

高级监控[1]是工业 4.0 背景下最重要的任务之一，高级监控利用传感器数据来实时监控生产系统中出现的各种问题，利于掌握生产线的生产状况，对于制造业确保高质量生产标准至关重要[2]。

传统的监控任务通过判断相关单变量参数是否超过预设的上下限定义异常，或者通过一些基于假设检验的方法定义异常。虽然该类方法在工业中仍然普遍存在，且有着较低的时间复杂度，但仅基于单变量的监测具有很大的局限性：例如会误判开关量导致的数值型指标的变化，或者生产物料输入的变化引起的变化，即无法捕捉复杂工业系统的多变量参数关联导致的异常，不符合高级监控的要求。

基于多变量机器学习的异常检测(Anomaly Detection, AD)技术现在被认为是有效的高级监控的首选解决方案，该类方法通过多变量联合考虑进行异常判断。在这类方法中，数据的不平衡性即缺乏可靠的异常标记数据，使得 AD 任务仅能依靠无监督的算法[3]。现有的无监督多变量异常检测方法多集中于复杂的神经网络，依赖大量数据进行长时间的训练，如果投入实际使用也需要实时的更新模型参数，消耗大量算力，而对于传统的机器学习方法，训练时间较短，算力开销小，但由于同一个模型往往在不同数据集上有很大的性能差异，模型的种类和超参数往往依赖于人工设计，使得对不同设备的异常检测显得异常繁琐，所以，需要一种自动化机器学习(Auto Machine Learning, AutoML)的方式去为不同分布的数据自动选择模型，提高效率。

因此，针对上述问题，我们提出了一种异常检测模型自动选择机制，该机制能够在大部分新数据集上自动选择最优的异常检测模型和超参数，实现 AD 任务的自动化。总的来说，我们的工作概括为以下三点：

定义了与数据集自身相关的统计特征以及一些任务相关的特征，称为元特征，并评估相关数据集在特定模型上的性能指标，这些特征和指标作为模型选择提供依据。

构建了一个深度神经网络，以数据集的元特征作为输入，以性能指标作为输出，通过定义特定的损失函数，训练模型，依据预测的模型性能指标进行模型选择，并根据选择到的模型训练和预测，应用到实际生产中。

在公开数据集上对比了以不同评价指标选出的模型的性能和排名，证明了自动化异常检测的一个可行性。

2. 相关工作

2.1. 异常检测

国内外研究者为多维度的无监督异常检测算法提供了很多的思路，大致可以分为以下几种类型：基于概率的方法，基于集成的方法，基于近邻的方法，基于深度学习的方法等。基于概率的方法通过统计学方法为数据建立一个概率模型，并认为正常数据应该出现在概率密度较高的区间，而异常数据应该出现在概率密度较低的区间。HBOS [4]是其中的代表性算法。此类方法的优点是运算速度快，缺点是依赖于特定的统计分布且不太适合高维数据。基于集成的方法旨在集成多种模型，采用投票、加权平均等方式来提升模型的整体性能。其中 IForest [5]是基于树模型的集成学习方法。此类方法的优点是能避免单一模型的过拟合问题，且具有良好的泛化能力，缺点是若集成的模型过多会导致训练和推理的时间增加。基于近邻的方法以距离作为度量来分离距离其他点较远的点，以此来作为异常数据。常见的方法如 KNN [6]等属于此类中的代表性算法。此类方法的优点是不会对数据分布做出任何假设，缺点是依赖距离的度量方式，当数据较为复杂或难以定义数据之间的距离时效果较差。基于深度学习的方法通过建立神经网络训练不断调整网络参数从而建立输入与输出之间的关系，通过重建误差来判断异常点。AE [7]等模型都能应用在异常检测任务中。此类方法的优点是模型准确率高，缺点是训练时间较长、可解释性差、容易过拟合等。此外，还有一类深度学习模型通过过去时刻的指标预测下一个时刻的指标，通过比较预测值和真实值的误差来判断异常，这一类算法通常用于时间序列的异常检测。

2.2. 自动化机器学习和元学习

为真实世界的复杂应用场景构建异常检测模型通常仍然严重依赖于人类的专业知识来微调超参数和设计模型结构，这些工作通常是耗时的，并且得到的解决方案仍然可能有次优的性能，所以需要一种自动化的方式，即自动化机器学习(Auto Machine Learning, AutoML)，来选择最佳模型。元学习(Meta-Learning) [8]是实现模型自动化选择的一种方式，可以包括下面的应用：超参数优化(Hyperparameter Optimization, HPO) [9]被认为是一种自动优化模型参数的手段，而神经结构搜索(Neural Architecture Search, NAS) [10]是神经网络结构自动化设计的一个有效的手段，可以通过强化学习和进化算法等来从数据中发现最佳的模型结构[11]。Auto-Sklearn [12]通过数据集的元特征来选择相似的数据集上表现最好的模型作为最终模型，AutoGluon [13]通过 Stacking 集成的方式结合多个基础模型提高模型的性能。

2.3. 常检测模型自动选择

把 AutoML 应用于 AD 的研究近年来逐渐成为研究热点。例如 Li [14]等人构建了一个用于建立统计学习异常检测模型自动化的框架，实现了非神经网络的参数优化。Li [15]等人用强化学习和 NAS 结合的方式自动构建神经网络异常检测模型的结构。Lai [16]等人构建了基于时序的自动化异常检测模型，并提供了可视化的操作界面。Kotlar [17]等人通过定义一些新的元特征作为模型选择的依据构建自动化异常检测模型。Zhao [18]等人通过把推荐领域协同过滤的方法用作模型推荐构建了异常检测自动化模型。

3. 模型设计

3.1. 问题描述

给定一个新数据集，或者有新的数据流来临时，自动化异常检测能够根据这一份新的数据集或者新

的批量的数据流给出一个可能的最佳异常检测模型，该模型对于该数据能够有较高的性能表现期望。现将自动化异常检测任务形式化定义如下：

现有若干不同领域历史数据集和标签 $\mathcal{D}_{train}=\{D_1,\ldots,D_n\}$，其中 $D_i=(X_i,y_i)$。

选定一些异常检测模型 $\mathcal{M}=\{M_1,\ldots,M_m\}$，这些模型能够检测出上述数据集的异常，并给出异常评分，结合标签能够给出性能指标。

有了不同领域不同分布的数据集和标签之后，我们可以计算上述异常检测模型在这些数据集上的性能，记为 $P\in R^{n\times m}$，其中 P_{ij} 代表一个数据集 D_i 在一个异常检测模型 M_j 上的性能指标，例如 *Pricision*、$F1$、*Recall*、*ROCAUC* 等。

自动化异常检测任务就是在上述先验性能的情况下，通过某种方式，当有新的数据集 $\mathcal{D}_{test}$ 来临时(该数据没有标签)，自动化异常检测能够自动选择一个异常检测模型 $M\in\mathcal{M}$，该模型可能是该数据集潜在的性能最佳的模型。

3.2. 算法步骤

前人对于自动化异常检测的工作主要集中在定义数据集的元特征、针对神经网络进行自动构建、通过较为简单的方式选择模型(例如协同过滤[1])集中于其中的某一方面。我们针对上述三方面做了改进，并提出了一个综合的方案：首先，我们加入了一些与异常检测相关的统计特征，让模型选择的过程更具领域性。其次，考虑工业生产中异常检测的时效性和算力要求，我们选择的模型主要集中在低时间复杂度的机器学习模型。最后，我们可以选用较为复杂的神经网络作为模型选择的方式。原因有以下四点：第一，模型预测的过程并不依赖于模型选择的过程，即高复杂度的模型选择过程不会拖慢模型的预测，我们选择的预测模型依然是时间复杂度较低的机器学习模型；第二，我们不关心模型选择的具体依据，但关心选择到的模型的性能；第三，我们可以找到大量公开数据集作为训练集；第四，依赖于特定的神经网络结构我们可以高效地查询潜在模型。我们给出了异常检测模型自动选择任务的形式化描述，并提出了 AutoAD (Auto Anomaly Detector)的方案。

我们提出的 AutoAD 有两个阶段：一个是训练阶段，也就是 Offline 的阶段，该阶段属于元学习的过程，主要是根据训练任务数据集本身的特征和数据集在不同模型上的性能表现等先验知识建立异常检测模型自动选择机制；二是预测阶段，即 Online 的阶段，该阶段能够运用训练阶段的异常检测模型自动选择机制给测试任务数据集一个可能的最佳模型，再用测试任务中的训练集(不带标签)训练该模型，并在测试任务中的测试集上评估性能。

3.2.1. 训练阶段

我们的训练阶段是一个元学习的过程，元学习是学习怎样学习模型的一个过程，通常是通过训练任务建立元模型，该元模型能够给出一个任务相关的最佳模型，然后通过测试任务评估该模型的性能。

一般情况下，元学习会在一系列历史任务学习经验，这些经验能让模型在一个给定的新的数据集上有类似先前相似任务的性能表现，这就需要我们定义两个任务的相似性。在机器学习中，任务之间的相似性是通过数据集的特征量化的，这些特征被称为元特征，它们通常捕获数据集数据分布的统计特性。为了获取这些数据集的元特征，我们参考[18]述数据集的一些特征，其中包括：

数据集相关的统计特征，例如偏度、峰度、标准熵等，刻画了数据集本身的特征，更多特征见表 1。

由于我们面向的是异常检测任务，加入一些异常检测相关的特征可以提高模型的性能。

除了表 1 中所示的相关统计特征外，我们还为数据集加入了一些异常检测相关的特征，这些特征可以表示数据集在异常检测这个任务上不同异常分布的一个差异，以此来强化数据集对于异常检测任务的一个差异表现，从而为模型选择提供更多的依据，下面解释我们加入的一些额外特征。

Table 1. Selected statistical features related to the dataset itself, as a feature representation of the dataset
表 1. 选取的与数据集自身相关的统计特征，作为数据集的特征表示

Name	Formula	Variants
Nr instances	n	$\frac{p}{n}$, $\log(n)$, $\log\left(\frac{n}{p}\right)$
Nr features	p	$\log(p)$, % categorical
Sample mean	μ	
Sample median	$\widetilde{X}$	
Sample var	σ^2	
Sample min	$\max_X$	
Sample max	$\min_X$	
Sample std	σ	
Percentile	P_i	$q1$, $q25$, $q75$, $q99$
Interquartile Range (IQR)	$q75-q25$	
Normalized mean	$\frac{\mu}{\max_X}$	
Normalized median	$\frac{\widetilde{X^X}}{\max X}$	
Sample range	$\max_X - \min_X$	
Sample Gini		
Median absolute deviation	$median\left(X-\widetilde{X}\right)$	
Average absolute deviation	$avg\left(X-\widetilde{X}\right)$	
Quantile Coefficient Dispersion	$\frac{(q75-q25)}{(q75+q25)}$	
Coefficient of variance		
Outlier outside 1% or 99%	%samples outlier 1% or 99%	
Outlier 3 STD	%samples outside 3σ	
Normal test		
k-th moments		5th to 10th moments
Skewness	Feature skewness	max, min, μ, σ, skewness, kurtosis
Kurtosis	$\frac{\mu_4}{\sigma^4}$	max, min, μ, σ, skewness, kurtosis
Correlation	ρ	max, min, μ, σ, skewness, kurtosis
Covariance	Cov	max, min, μ, σ, skewness, kurtosis
Sparsity	$\frac{\#Unique\ values}{n}$	max, min, μ, σ, skewness, kurtosis
ANOVA p-value	p^n_{ANOVA}	max, min, μ, σ, skewness, kurtosis
Coeff of variation	$\frac{\sigma_x}{\mu_x}$	
Norm. entropy	$\frac{H(X)}{\log_2 n}$	max, min, μ, σ

基于孤立森林的相关特征：孤立森林(IForest)是一种基于树的集成方法，它将数据点随机分割成子空

间，递归地构建一棵二叉树，并使用树的深度来确定每个数据点的异常得分。具体来说，IForest 使用采样的数据建立一个基树集合，以随机选择的特征为节点进行分割，持续直到叶子节点只包含一个样本或达到预定的最大深度，较浅的叶子节点意为拆分时更容易被“隔离”，意为较高的异常分数，因此，异常点更接近于树深较小的根部。对于 IForest，我们使用树的深度和每棵树的叶子数量以及每个基础树的特征重要性来作为统计特征。具体来说，我们使用了基树的以下信息：

树的深度：最小，最大，平均值，偏度和峰度。

叶子的数量：最小，最大，平均值，偏度和峰度。

基树特征重要性的平均值：最小，最大，平均值，偏度和峰度。

基树特征重要性的最大值：最小，最大，平均值，偏度和峰度。

基于 HBOS 的相关特征：HBOS 假设数据集的每个维度是独立的，它在每个特征上建立一个直方图来计算密度，鉴于有 n 个样本和 d 个特征，对于每个直方图，HBOS 使用所有 n 个样本估计样本密度。直观地说，样本的异常得分被定义为反密度的对数之和，落在高密度区域的样本更有可能是正常点，反之则为异常点。我们使用了如下特征：

每个直方图的平均值(每个特征)：最小、最大、平均值、标准差、偏度和峰度。

每个直方图的最大值(每个特征)：最小、最大、平均值、标准差、偏度和峰度。

基于 LODA 的相关特征：LODA 是一种基于集合的快速异常检测算法，它与 HBOS 的想法相似，但与简单地汇总所有独立直方图的 HBOS 不同，LODA 扩展了基于直方图的模型，生成 k 个随机投影向量，将数据压缩到一维空间以构建直方图。我们将以下信息作为元特征的一部分纳入其中：

每个随机投影的平均值(每个特征)：最小，最大，平均值，标准差，偏度和峰度。

每个随机投影的最大值(每个特征)：最小、最大、平均值、标准差、偏度和峰度。

基于 PCA 的相关特征：PCA 旨在通过主成分分析将样本投射到较低的维度来量化它们的异常值。由于正常样本的数量远远大于异常值的数量，因此所确定的投影矩阵主要适用于正常样本，正常样本的重建误差比异常值样本的重建误差要小，可以用来衡量样本的离群性。对于 PCA，我们将以下信息纳入元特征：

前三个主成分上的解释方差率：它为前 3 个主成分捕获的方差的百分比。

奇异值：SVD 过程中产生的前 3 个奇异值。

我们通过计算训练任务数据集 D_i 的上述相关特征，组成数据集的特征向量 F_i，计算不同数据集上的特征向量组成 $F=\oint\left(\{X_1,\cdots,X_n\}\right)\in R^{n\times d}$ 。

同时，为了为模型选择提供依据，我们要评估不同数据集在不同异常检测模型上的性能，这里的模型同时包括了异常检测基准模型以及不同超参数组合的变体模型，这样可以实现在模型选择的过程中同时调整了超参数。我们评估训练任务数据集 D_i 在模型 M_j 上的性能表现 P_{ij}，评估不同数据集在多个异常检测模型上的性能指标组成 $P\in R^{n\times m}$ 。具体的模型以及超参数组合见实验结果与分析部分表 2。

至此，我们获得了训练任务上的两部分先验知识 F 和 P，我们通过这两部分先验知识构建异常检测自动选择机制的训练集：把 F 作为我们的输入特征，把 P 作为模型的输出。图 1 展示了上述的整体流程。

由于模型的输出是输入数据集在不同模型上的一个性能指标的期望分布，我们希望模型的输出还原这个数据集的结果的真实分布，但这一条件可能太过苛刻：我们希望的是在数值相差可以容忍的范围内，大小顺序接近这个真实分布的大小顺序，并且输出结果排名和原始分布排名相同或者相近的情况给予更多的奖励，所以我们选择了 NDCG 作为我们训练模型的目标函数。

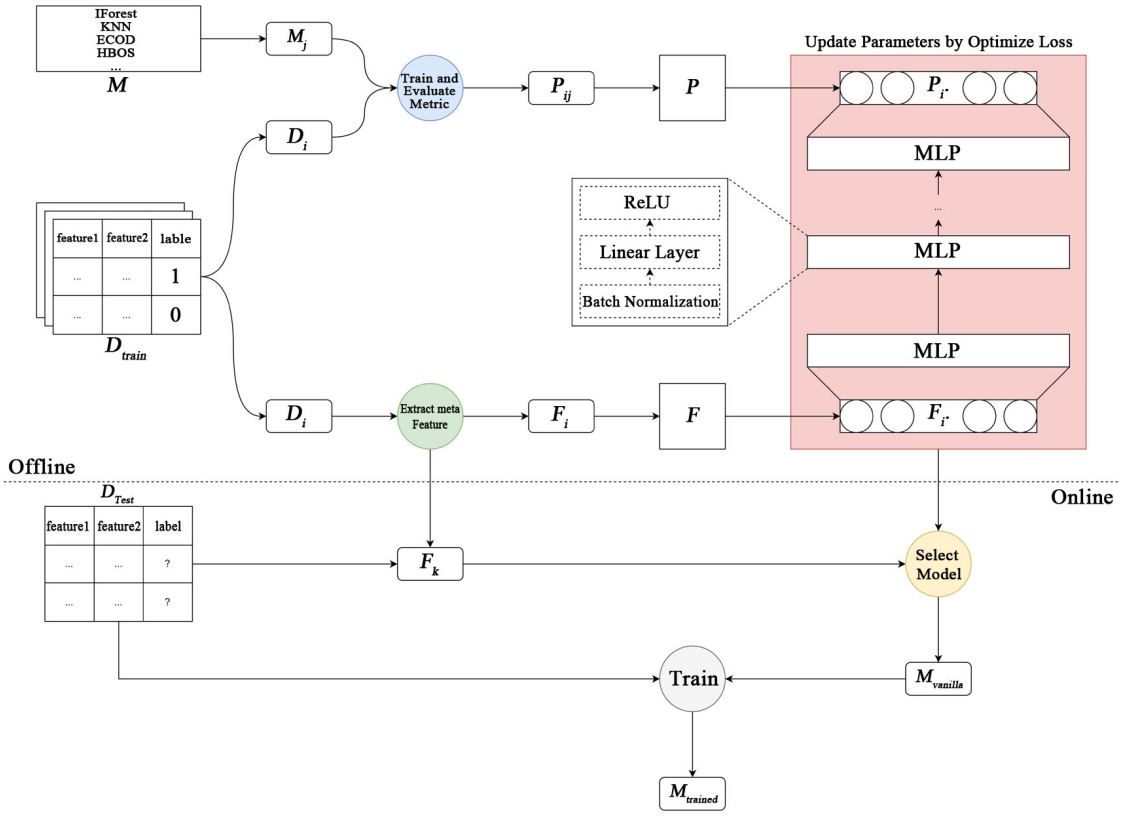

Figure 1. The overall process of training and testing the model, where the green part represents the process of computing the meta-feature F of the dataset, the blue part represents the process of evaluating the performance of different datasets on different models, the red part represents the process of updating the parameters of the AutoAD neural network through F and P, and the yellow part represents the process of selecting the model based on the trained AutoAD. The gray part represents the process of training the selected model on D_{test} to get the final model.

图 1. 训练模型和测试模型的整体流程，其中绿色部分表示计算数据集元特征 F 的过程，蓝色部分表示评估不同数据集在不同模型上的性能表现的过程，红色部分表示通过F和P进行 AutoAD 神经网络参数更新的过程，黄色部分表示依据训练好的 AutoAD 进行模型选择的过程，灰色部分表示把选出的模型在 D_{test} 上进行训练得到最终的模型。

NDCG 原本是用来衡量搜索结果的评价指标，搜索到的结果的相关性越高且越靠前，则 NDCG 越高。

$$DCG_p = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

其中 *rel* 就是搜索到的结果与搜索词的相关性，i 为这个结果在搜索列表中的排名。因为不同的搜索结果的相关性 *rel* 差别较大，为了让结果可比，对 DCG 做了归一化。

$$IDCG_p = \sum_{i=1}^{|REL_p|} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

$$NDCG_p = \frac{DCG_p}{IDCG_p}$$

其中 *REL* 为文档库中相关性最高的文档集合。

应用到我们的应用场景里，相关性 *rel* 即模型输出的值的真实值，*REL* 为所有真实值从大到小排名后的集合，而 i 就为模型输出值的排名。模型输出的结果用于判断排名，所以可以把 NDCG 用于我们的选择任务中。然而 NDCG 有一个问题就是不可导，所以我们采用了[19]的方法进行了一些优化，并把最终

优化后的版本作为我们的损失函数。

3.2.2. 测试阶段

对于测试阶段，给定一个新的不带任何标签的数据集，先计算该数据集对应的元特征，然后把元特征输入到上述的模型中，模型会计算该数据集在各个给定的模型上的评估指标，做一个排名之后给出最佳模型或者 Top-k 个模型。对于给出的模型，我们再用数据集进行训练得到最后的异常检测模型，最终用该模型进行预测。由于选择的模型中包括了基准模型和基准模型不同超参数的变体，这样就完成了整个模型选择和调优的过程以及最终模型推理的过程。

4. 实验结果及分析

我们评估了 AutoAD 对于潜在高性能指标异常检测模型的选择能力，并与候选的其他基准模型进行了性能上的比较。

4.1. 实验设置

我们准备了与 AutoAD 进行对比的候选基准模型以及这些基准模型调整超参数后的模型，总计 66 个，相关模型和参数设置见表 2，在基准模型中加入调整过超参数后的模型可以让我们的异常检测模型自动化选择的过程中考虑这些不同超参的模型，实现了在选择模型的同时调整超参数。在训练模型时，我们采用正态分布的数初始化模型参数，我们设置了学习率为 1e−3，设置了 100 个 epoch，优化器采用 Adam，批量大小设置为 32。

Table 2. Candidate benchmark models and a combination of hyperparameters, where the hyperparameters are Cartesian products of relations
表 2. 候选基准模型以及超参数组合，其中超参数为笛卡尔积的关系

Model	Hyper Param #1	Hyper Param #2	Hyper Param #3
LOF	neighbors：[5, 10, 20, 30, 50]	Metric: [‘manhattan’, ‘euclidean’, ‘minkowski’]	None
COF	neighbors：[5, 10, 20, 30, 50]	None	None
HBOS	bins：[5, 10, 20]	Alpha: [0.05, 0.1, 0.2]	tol: [0.1, 0.5, 0.9]
KNN	neighbors：[2, 5, 10, 20]	None	None
SOD	neighbors：[20, 30, 50]	None	None
GMM	components：[1, 3, 5]	None	None
LODA	bins：[5, 10, 20]	None	None
IForest	estimators：[50, 100, 200]	None	None

4.2. 数据集

在训练阶段，我们使用了 ADBench [20]中的 57 个不同领域的异常数据集，这些数据集来自不同的领域有着不同的分布，因为我们自动化模型选择的能力较大程度依赖于训练时数据分布的多样性，所以为了增强数据的多样性，我们将这些数据集做了划分，最终形成了 1404 个小数据集作为我们的训练任务，我们在这些小数据集上训练了我们的 AutoAD。

在测试阶段，我们使用了开源可用的数据集作为测试任务，每一个测试任务中都包含了用于训练选择到的模型的训练集和用于评估指标的测试集。我们在表 3 中总结了它们的特点。

Table 3. An overview of the datasets for the test tasks, where each test task contains the training set and the test
表 3. 测试任务的数据集概述，其中的每一个测试任务都包含了训练集和测试

Dataset	#Train	#Test	Dim	Channel	%Outlier
SWaT	496800	449919	51	1	11.98
WADI	1048571	172801	123	1	5.99
SMD	708405	708420	38	4	4.16
SMAP	135183	427617	25	55	13.13
MSL	58317	73729	55	3	10.72

4.3. 评价指标

评价指标异常检测我们使用 ROCAUC 得分来评价所有模型的检测性能。

ROCAUC 的计算公式可以用以下数学式表示：

$$ROCAUC = \int_0^1 TPR\left(FPR^{-1}(t)\right)dt$$

其中，TPR 表示真正例率(True Positive Rate)，FPR 表示假正例率(False Positive Rate)，FPR^{-1} 表示 FPR 的逆函数，t 表示在 ROC 曲线上的阈值。该公式表示的是 ROC 曲线下的面积，可以通过计算 ROC 曲线的数值近似求解，其中 ROC 曲线的坐标点可以通过不同的阈值计算得到。

4.4. 基准模型

LOF (Local Outlier Factor)：LOF 算法主要用于识别数据集中的离群点，即与大多数数据点相比具有不同行为模式的数据点。

COF (Cluster-based Local Outlier Factor)：是 LOF 的一个扩展，旨在进一步提高 LOF 算法对于聚类结构的敏感性。

HBOS (Histogram-Based Outlier Score)：模型利用直方图分布对数据特征进行建模，通过异常得分度量数据点相对于正常数据的偏离程度。

KNN (K-Nearest Neighbors)：对每个数据点，找到离其最近的 k 个邻居，并通过邻居之间的距离来计算该点的异常得分。

SOD (Subspace Outlier Detection)：一种用于检测多维数据中子空间离群点的算法。它的目标是在高维数据中发现子空间中的异常点，即与大多数数据点在某个子空间中行为不同的点。

GMM (Gaussian Mixture Model) ：是一种常见的异常检测算法，其基本思想是使用高斯混合模型对数据进行建模，然后利用模型来估计数据点是否异常。GMM 假设数据是由若干个高斯分布组成，每个高斯分布对应一个簇，然后通过最大似然估计来确定每个高斯分布的参数，包括均值和协方差矩阵。在预测时，GMM 算法计算每个数据点的概率密度值，并将概率密度值低于某个阈值的点视为异常。

LODA (Lightweight On-line Detector of Anomalies)：是一种轻量级的异常检测算法，它主要通过对数据进行分解，然后比较分解后的结果来检测异常。LODA 将数据分解为若干个子空间，每个子空间对应一个 LODA 模型。LODA 模型基于子空间的随机投影来估计数据点的概率密度值，然后将所有子空间的结果进行加权平均，得到最终的异常得分。与其他异常检测算法相比，LODA 具有较低的计算复杂度和较高的可扩展性。

IForest (Isolation Forest)：将数据点随机分割成子空间，递归地构建一棵二叉树，并使用树的深度来确定每个数据点的异常得分。

ECOD (Ensemble of Cluster-Based Outlier Detection)：模型通过聚类技术将数据分为多个子集，然后构建子集之间的异常度，最后集成这些异常度以识别异常数据点。

COPOD (Copula-Based Outlier Detection)：一种基于统计学 copula 的异常检测方法。

PCA (Principal Component Analysis)：使用主成分分析来对数据进行降维，并通过计算残差来确定数据点的异常得分。

4.5. 结果分析

4.5.1. AutoAD 选出的模型与候选基准模型的 ROCAUC 及其排名的对比

Table 4. Comparison of the ROCAUC metrics of AutoAD with candidate benchmark models, where the numbers in parentheses are the rankings of the different models on a particular channel of the dataset, and the average rankings were tallied for each dataset, with higher ROCAUC metrics and smaller rankings, with AutoAD's prominence highlighted in bold.
表 4. AutoAD 与候选基准模型的 ROCAUC 指标对比，其中括号内的数字是不同模型在数据集的某个频道上的排名，每个数据集都统计了平均排名，ROCAUC 指标越高越好，排名越小越好，AutoAD 的突出点被粗体标出

Dataset	Channel	AutoAD	LOF	COF	HBOS	KNN	SOD	GMM	LODA	IForest	ECOD	COPOD	PCA
SMD	machine-1-4	0.5131(3)	0.5346(2)	0.5595(1)	0.5071(4)	0.4932(5)	0.4838(6)	0.3376(9)	0.4447(7)	0.3658(8)	0.2608(11)	0.2408(12)	0.2777(10)
	machine-1-6	0.6126(2)	0.5427(5)	0.5037(7)	0.6067(3)	0.6161(1)	0.525(6)	0.5883(4)	0.3084(10)	0.4683(8)	0.0(11)	0.0(11)	0.4272(9)
	machine-1-8	0.5883(2)	0.4797(7)	0.4833(6)	0.4859(5)	0.5831(3)	0.577(4)	0.6592(1)	0.3647(9)	0.3444(11)	0.3367(12)	0.4518(8)	0.3574(10)
	machine-2-3	**0.5222(1)**	0.2934(5)	0.415(3)	0.1729(8)	0.3779(4)	0.4547(2)	0.2902(6)	0.2212(7)	0.1414(9)	0.0(11)	0.0(11)	0.1352(10)
	machine-2-6	**0.5613(1)**	0.4326(5)	0.493(4)	0.2337(11)	0.5253(2)	0.5122(3)	0.4275(6)	0.3103(8)	0.2136(12)	0.299(9)	0.2409(10)	0.3456(7)
	machine-3-3	**0.5095(1)**	0.4744(3)	0.4722(4)	0.2952(9)	0.4852(2)	0.4459(6)	0.4717(5)	0.396(7)	0.3924(8)	0.0(11)	0.0(11)	0.2553(10)
	machine-3-6	0.5507(3)	0.5472(4)	0.5377(5)	0.378(10)	0.5828(1)	0.5534(2)	0.5299(6)	0.4089(9)	0.3294(12)	0.4758(7)	0.3559(11)	0.445(8)
	average	**1.86**	4.43	4.29	7.14	2.57	4.14	5.29	8.14	9.71	10.5	10.79	9.14
MSL	M-4	0.7316(3)	0.5778(8)	0.6968(4)	0.7909(1)	0.7464(2)	0.6223(6)	0.5285(11)	0.3775(12)	0.6116(7)	0.5741(9)	0.6308(5)	0.5588(10)
	M-5	0.7449(2)	0.6159(4)	0.586(5)	0.6647(3)	0.7451(1)	0.5859(6)	0.4136(9)	0.4807(8)	0.5595(7)	0.0(11)	0.0(11)	0.4068(10)
	P-14	0.7541(2)	0.7506(5)	0.7498(6)	0.4889(9)	0.7541(2)	0.6406(7)	0.7541(2)	0.2745(10)	0.4894(8)	0.0(11)	0.0(11)	0.7541(2)
	average	2.5	5.67	5.0	4.33	1.83	6.33	7.5	10.0	7.33	10.67	9.33	7.5
SMAP	A-4	0.821(2)	0.7957(3)	0.4593(10)	0.2394(12)	0.8297(1)	0.3145(11)	0.706(6)	0.6516(8)	0.7293(4)	0.7(7)	0.5855(9)	0.707(5)
	B-1	0.4995(2)	0.4975(4)	0.4975(4)	0.4918(8)	0.4994(3)	0.5032(1)	0.4918(8)	0.4919(6)	0.4918(8)	0.0(11)	0.0(11)	0.4918(8)
	D-11	**0.7856(1)**	0.7774(2)	0.4832(6)	0.2822(10)	0.7758(3)	0.5009(5)	0.2837(8)	0.4412(7)	0.5516(4)	0.0(11)	0.0(11)	0.2837(8)
	D-2	0.5008(3)	0.4982(6)	0.5003(5)	0.4969(9)	0.5004(4)	0.5042(1)	0.4973(7)	0.504(2)	0.4973(7)	0.4967(11)	0.4967(11)	0.4967(11)
	D-3	0.9733(2)	0.9755(1)	0.5465(9)	0.7876(7)	0.9722(3)	0.6232(8)	0.9685(5)	0.2343(10)	0.8175(6)	0.0(11)	0.0(11)	0.9686(4)
	D-4	**0.9998(1)**	0.9846(5)	0.938(6)	0.7947(9)	0.9995(2)	0.7985(8)	0.9984(3)	0.0695(10)	0.8741(7)	0.0(11)	0.0(11)	0.9984(3)
	D-8	**0.4994(1)**	0.4974(3)	0.4974(3)	0.495(8)	0.4985(2)	0.495(8)	0.495(8)	0.4951(5)	0.495(8)	0.0(11)	0.0(11)	0.495(8)
	D-9	**0.4995(1)**	0.4962(4)	0.4962(4)	0.4922(8)	0.4984(2)	0.4922(8)	0.4922(8)	0.4967(3)	0.4922(8)	0.0(11)	0.0(11)	0.4922(8)
	E-1	**0.7164(1)**	0.7138(2)	0.5(11)	0.6698(5)	0.7138(2)	0.6969(4)	0.6306(7)	0.3336(12)	0.6591(6)	0.627(9)	0.627(9)	0.6271(8)
	E-12	**0.6884(1)**	0.6884(1)	0.5015(8)	0.4682(9)	0.6857(3)	0.4641(10)	0.3248(11)	0.6741(4)	0.6209(5)	0.6161(6)	0.6161(6)	0.32(12)
	E-7	**0.6241(1)**	0.6002(3)	0.56(12)	0.5678(8)	0.6194(2)	0.5603(11)	0.5835(5)	0.5983(4)	0.5726(7)	0.5674(9)	0.567(10)	0.5826(6)
	F-2	0.7881(2)	0.788(3)	0.5135(8)	0.7483(4)	0.7893(1)	0.4907(9)	0.724(6)	0.318(10)	0.6064(7)	0.0(11)	0.0(11)	0.7244(5)
	G-1	0.6197(3)	0.3208(7)	0.211(12)	0.4744(5)	0.7181(2)	0.2627(9)	0.2662(8)	0.7574(1)	0.5183(4)	0.2575(10)	0.3681(6)	0.2406(11)
	G-4	0.5321(2)	0.5321(2)	0.5321(2)	0.4775(7)	0.5321(2)	0.4694(8)	0.4776(5)	0.4123(10)	0.4693(9)	0.0(11)	0.0(11)	0.4776(5)
	P-1	0.51(3)	0.4236(8)	0.3169(10)	0.482(6)	0.4458(7)	0.3447(9)	0.4848(5)	0.5438(1)	0.5097(4)	0.0(11)	0.0(11)	0.5371(2)
	P-4	0.4998(2)	0.4986(4)	0.496(5)	0.4874(8)	0.4995(3)	0.4886(6)	0.4874(8)	0.511(1)	0.4874(8)	0.0(11)	0.0(11)	0.4874(8)
	S-1	0.7432(2)	0.6969(5)	0.3766(10)	0.7171(3)	0.7457(1)	0.3461(12)	0.6545(8)	0.3604(11)	0.5803(9)	0.6995(4)	0.6648(6)	0.6552(7)
	average	**1.82**	3.88	7.47	7.47	2.59	7.53	7.0	6.18	6.62	10.12	10.18	7.15
SWaT	/	0.8162(8)	0.3757(9)	0.0(11)	0.8437(4)	0.0(11)	0.2594(10)	0.8254(6)	0.8571(1)	0.8393(5)	0.8556(2)	0.8528(3)	0.8164(7)
	average	8.0	9.0	11.5	4.0	11.5	10.0	6.0	1.0	5.0	2.0	3.0	7.0
WADI	/	0.4733(9)	0.477(7)	0.6182(3)	0.7543(1)	0.4757(8)	0.5735(4)	0.4863(6)	0.42(10)	0.7208(2)	0.0(11)	0.0(11)	0.5031(5)
	average	9.0	7.0	3.0	1.0	8.0	4.0	6.0	10.0	2.0	11.5	11.5	5.0

我们比较了 AutoAD 选择出的模型与其他主流的异常检测模型的 ROCAUC 性能，结果如表 4 所示。该表格展示了不同异常检测算法在五个数据集(SMD、MSL、SMAP、SWaT、WADI)的不同频道上的 ROCAUC 结果，表格中的模型是我们训练 AutoAD 时的候选基准异常检测模型(保留默认超参数的模型)，其中括号内表示模型在数据集某个频道上的排名，对于 AutoAD 表现出彩的地方已经用粗体标出。从表

格中可以看出，在 SMD 数据集上，AutoAD 选出的模型在大部分频道上都表现出了最优和次优的水平，并且平均排名也超越了所有的候选基准异常检测算法；在 MSL 数据集上，模型取得了平均第二的水平；在 SMAP 数据集上，AutoAD 选出的模型在大部分频道上都表现出了最优和次优的水平，并且平均排名也超越了所有的候选基准异常检测算法；在 SWaT 数据集上，虽然 AutoAD 选出的模型排名不高，但是 ROCAUC 指标在排名比它靠前的模型中差别不大，可能是由于该数据集在不同模型上性能表现相近导致的；而在 WADI 数据集上，AutoAD 的表现就不是很好，可能是由于这个数据集的分布与训练 AutoAD 时的训练任务分布差异较大导致的。但总体来看，AutoAD 还是实现了异常检测模型的高效自动选择，并在大多数情况下都有较好的表现。

4.5.2. AutoAD 以不同评价指标为目的选出的模型对比候选基准模型的排名对比

为了探究以不同检测评价指标为目的训练的 AutoAD 能不能也有像上述以 ROCAUC 为目的训练的 AutoAD 相似的异常检测模型选择能力，我们训练了针对不同评价指标的 AutoAD 变体：AutoAD_Average_Precision、AutoAD_F1、AutoAD_Precision、AutoAD_Recall，即分别以 Average Precision、F1、Precision、Recall 为评价指标训练的 AutoAD，比较了这四种 AutoAD 与其他候选基准模型对于数据集评价指标的平均排名情况，如图 2 所示(因为 SWaT 和 WADI 只有一个频道，被忽略)。从结果可以看出，采用相同训练方式不同评价指标方式训练的 AutoAD 对于模型选择在不同的评价指标下都超越了候选基准模型的平均排名，所以以不同的选择目标训练 AutoAD 依然可以达到选择较优模型的效果。

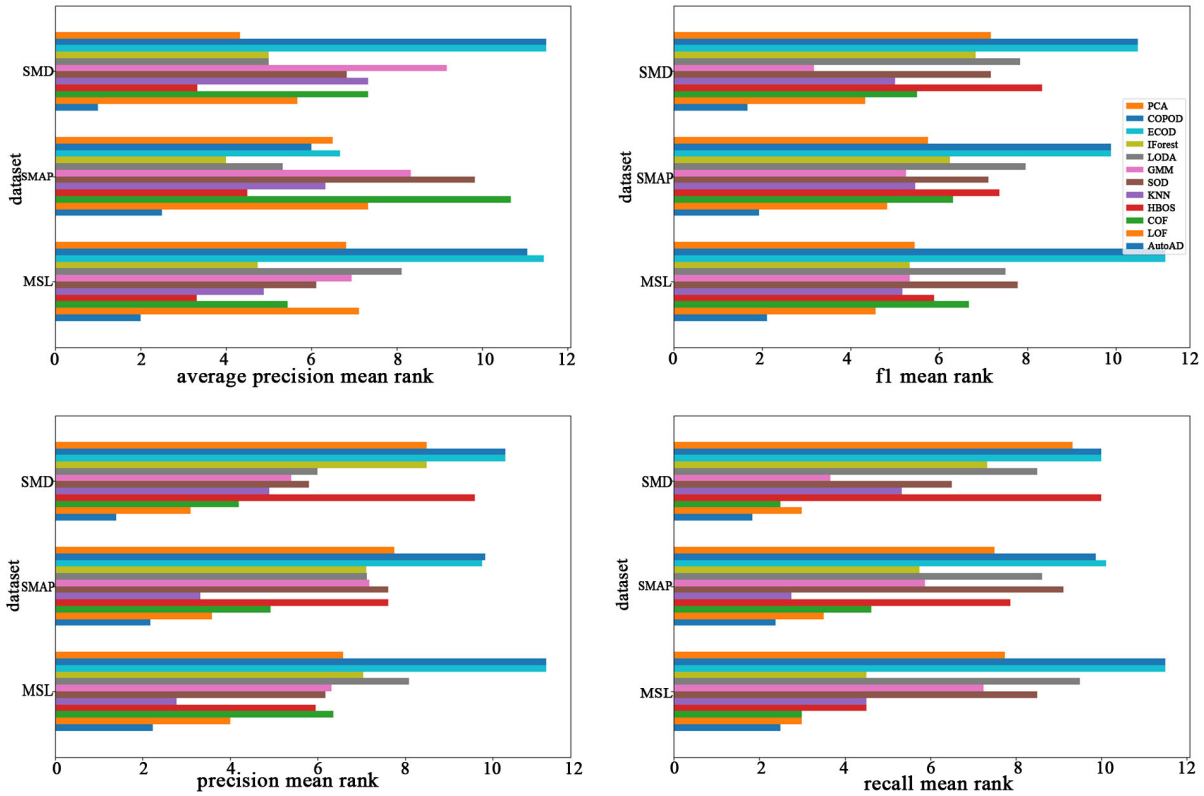

Figure 2. AutoAD's models trained and selected with different evaluation metrics objectives are compared with the candidate benchmark models on different evaluation metrics, with the smaller rankings being the smaller the ranking, the better

图 2. AutoAD 以不同评价指标目标训练并选出的模型与候选基准模型在不同评价指标上的对比，排名越小越好

4.5.3. AutoAD 与其他异常检测自动化方法的对比

为了对比 AutoAD 与其他自动化异常检测 MetaOD 的性能，我们评估了两者和候选基准模型在 Average Precision 评价指标上的性能(因为 MetaOD 只有根据这个指标进行建模)，排名对比如图 3 所示。从图中可以看出 AutoAD 在大部分数据集上的平均排名都超越了候选基准模型和 MetaOD，在 SWaT 数据集上 AutoAD 的平均排名与 MetaOD 拉开了很大差距，而在 WADI 数据集上两者差距不大，但是 AutoAD 平均排名没有超过基准模型 LODA 和 IForest。

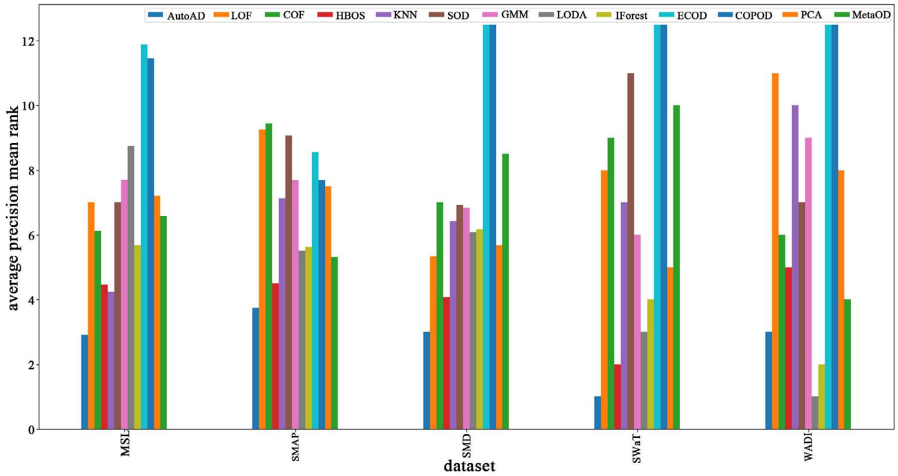

Figure 3. Comparison of the average rankings of AutoAD, MetaOD, and other candidate models on the Average Precision metric, with smaller rankings being preferred
图 3. AutoAD、MetaOD 以及其他候选模型在 Average Precision 指标上的平均排名对比，排名越小越好

4.5.4. AutoAD 选择模型时间与评估完候选基准模型的时间对比

为了说明 AutoAD 在自动化异常检测时间上的有效性，我们比较了采用 AutoAD 选择模型的时间和统计计算完所有 11 个基准模型并且评估所需的时间做了一个对比，结果如表 5 所示。表中说明了用于统计时间的数据子集的数量和维度，以及 AutoAD 选择模型花费的时间和评估完所有候选基准模型所花费的时间。从表中可以看出 AutoAD 选择模型的时间在数据规模较大且维度较多的时候(SWaT 和 WADI)，对比评估所有候选基准模型有巨大的时间优势；在数据集规模较小(MSL 和 SMAP)或者维度较低(SMD)的情况下，AutoAD 选择模型对比评估所有候选基准模型仍然有巨大优势，不过评估所有候选基准模型所花的时间有所减少。从上述分析可以看出 AutoAD 在选择模型方面的节约时间的有效性，并且 AutoAD 的搜索空间是基准模型超参数扩展之后的空间(总共 66 个模型和超参数组合)，相比于随机搜索和全搜索，AutoAD 具有时间消耗上的巨大优势。

Table 5. AutoAD's time to select a model versus the time spent selecting a model after evaluating all benchmark models, which also includes the size and dimensionality of the data subset
表 5. AutoAD 选择模型的时间与评估完所有基准模型后选择模型所花费的时间的对比，其中还包括了数据子集的大小和维度

	#Test	Dim	AutoAD	Eval All
SMD	28479	38	0.044099	39.14166
MSL	2158	55	0.015389	0.177447
SMAP	2880	25	0.00468	0.847208
SWaT	9900	51	0.688104	130.483889
WADI	15692	127	2.713834	240.400134

4.6. 消融实验

额外的异常检测相关特征对结果的影响

为了探究 AutoAD 引入的额外异常检测相关特征是否影响最终的结果，我们对比了 AutoAD 及其变体(以不同评价指标为目的训练的 AutoAD)和相应的去除额外特征的 AutoAD-Feature 在不同数据集上的表现，结果如图 4 所示。从结果可以看出，在所有数据集上，AutoAD 相比 AutoAD-Feature 都表现得更好，因此可以认为相关异常检测特征对 AutoAD 算法在这些数据集上有正面影响，更多的相关特征能让模型发现更多的差异性，进而提升 AutoAD 选择模型的性能。

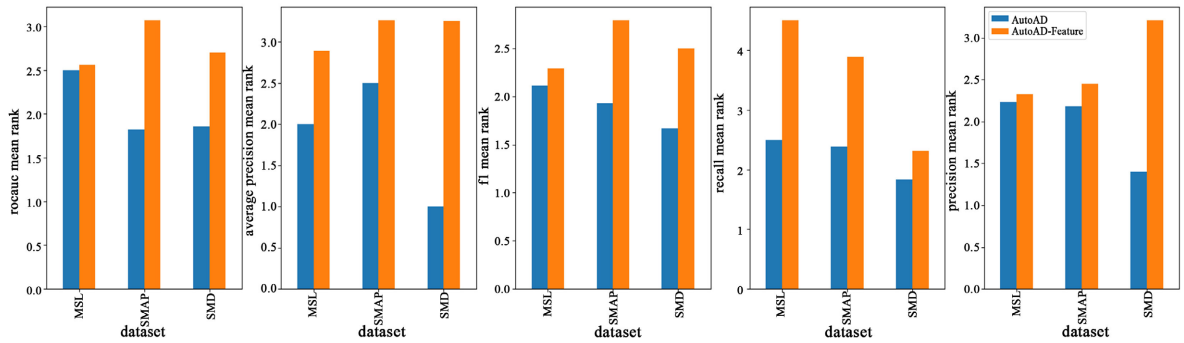

Figure 4. AutoAD vs. AutoAD-Feature with additional features removed in Average Precision metric ranking, smaller is better

图 4. AutoAD 与删除额外特征后的 AutoAD-Feature 在 Average Precision 指标排名对比，排名越小越好

7. 结语

我们提出了一个基于深度学习的用于无监异常检测的模型选择方法 AutoAD，该方法基于元学习的基本思想，根据不同分布的历史数据集上的指标评估建模。对于给定一个新的任务，它根据模型在类似历史数据集上的表现来选择一个模型。为了有效地捕捉任务的相似性，我们设计了针对该问题的元特征，包括了广泛应用的统计特征以及一些异常检测相关的元特征。AutoAD 是完全无监督的，在测试时不需要模型评估，此外，只有模型选择时用到了深度网络，在模型估计之前产生相对较小的选择时间开销。在测试任务上的实验结果表明，AutoAD 比直接使用一些流行的模型提高了性能，并且省去了模型选择和调优的时间消耗，具有实际意义。

参考文献

[1] Cheng, Y., Zhu, H., Wu, J. and Shao, X. (2019) Machine Health Monitoring Using Adaptive Kernel Spectral Clustering and Deep Long Short-Term Memory Recurrent Neural Networks. *IEEE Transactions on Industrial Informatics*, **15**, 987-997. https://doi.org/10.1109/TII.2018.2866549

[2] Vogel-Heuser, B. and Hess, D. (2016) Guest Editorial Industry 4.0-Prerequisites and Visions. *IEEE Transactions on Automation Science and Engineering*, **13**, 411-413. https://doi.org/10.1109/TASE.2016.2523639

[3] Carletti, M., Masiero, C., Beghi, A. and Susto, G.A. (2019) Explainable Machine Learning in Industry 4.0: Evaluating Feature Importance in Anomaly Detection to Enable Root Cause Analysis. 2019 *IEEE International Conference on Systems, Man and Cybernetics* (*SMC*), Bari, 6-9 October 2019, 21-26. https://doi.org/10.1109/SMC.2019.8913901

[4] Goldstein, M. and Dengel, A. (2012) Histogram-Based Outlier Score (HBOS): A Fast Unsupervised Anomaly Detection Algorithm. *KI*-2012: *Poster and Demo Track*, Vol. 1, 59-63.

[5] Liu, F.T., Ting, K.M. and Zhou, Z.-H. (2008) Isolation Forest. 2008 8*th IEEE International Conference on Data Mining*, Pisa, 15-19 December 2008, 413-422. https://doi.org/10.1109/ICDM.2008.17

[6] Abid, A., Kachouri, A., Ben FradjGuiloufi, A., Mahfoudhi, A., Nasri, N. and Abid, M. (2015) Centralized KNN Ano-

maly Detector for WSN. 2015 *IEEE* 12*th International Multi-Conference on Systems*, *Signals & Devices* (*SSD*15), Yasmine Hammamet, 19-22 March 2018, 1-4. https://doi.org/10.1109/SSD.2015.7348091

[7] Chen, Z., Yeo, C.K., Lee, B.S. and Lau, C.T. (2018) Autoencoder-Based Network Anomaly Detection. 2018 *Wireless Telecommunications Symposium* (*WTS*), Phoenix, 17-20 April 2018, 1-5. https://doi.org/10.1109/WTS.2018.8363930

[8] Hospedales, T., Antoniou, A., Micaelli, P. and Storkey, A. (2022) Meta-Learning in Neural Networks: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **44**, 5149-5169.

[9] Hutter, F., Kotthoff, L. and Vanschoren, J. (2019) Automated Machine Learning: Methods, Systems, Challenges. Springer Nature, Berlin, 219. https://doi.org/10.1007/978-3-030-05318-5

[10] Zoph, B. and Le, Q.V. (2017) Neural Architecture Search with Reinforcement Learning.

[11] Pham, H., Guan, M., Zoph, B., Le, Q. and Dean, J. (2018) Efficient Neural Architecture Search via Parameters Sharing. *Proceedings of the* 35*th International Conference on Machine Learning*, Stockholm, 10-15 July 2018, 4095-4104.

[12] Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M. and Hutter, F. (2015) Efficient and Robust Automated Machine Learning. *NIPS* 2015, Montreal, 11-12 December 2015, 28.

[13] Erickson, N., Mueller, J., Shirkov, A., Zhang, H., Larroy, P., Li, M. and Smola, A. (2020) AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data.

[14] Li, Y., Zha, D., Venugopal, P.K., Zou, N. and Hu, X. (2020) PyODDS: An End-to-End Outlier Detection System with Automated Machine Learning. *WWW*'20: *Companion Proceedings of the Web Conference*, Taipei, 20-24 April 2020, 153-157. https://doi.org/10.1145/3366424.3383530

[15] Li, Y., Chen, Z., Zha, D., Zhou, K., Jin, H., Chen, H. and Hu, X. (2020) AutoOD: Automated Outlier Detection via Curiosity-Guided Search and Self-Imitation Learning.

[16] Lai, K.-H., Zha, D., Wang, G., Xu, J., Zhao, Y., Kumar, D., Chen, Y., Zumkhawaka, P., Wan, M., Martinez, D. and Hu, X. (2021) TODS: An Automated Time Series Outlier Detection System. *Proceedings of the AAAI Conference on Artificial Intelligence*, **35**, 16060-16062. https://doi.org/10.1609/aaai.v35i18.18012

[17] Kotlar, M., Punt, M., Radivojević, Z., Cvetanović, M. and Milutinović, V. (2021) Novel Meta-Features for Automated Machine Learning Model Selection in Anomaly Detection. *IEEE Access*, **9**, 89675-89687. https://doi.org/10.1109/ACCESS.2021.3090936

[18] Zhao, Y., Rossi, R.A. and Akoglu, L. (2021) Automating Outlier Detection via Meta-Learning.

[19] Qin, T., Liu, T.Y. and Li, H. (2010) A General Approximation Framework for Direct Optimization of Information Retrieval Measures. *Information Retrieval*, **13**, 375-397. https://doi.org/10.1007/s10791-009-9124-x

[20] Han, S., Hu, X., Huang, H., Jiang, M. and Zhao, Y. (2022) Adbench: Anomaly Detection Benchmark. *Advances in Neural Information Processing Systems* 35: *Annual Conference on Neural Information Processing Systems* 2022, *NeurIPS* 2022, New Orleans, 28 November-9 December 2022, 32142-32159. https://doi.org/10.2139/ssrn.4266498