

复现论文题目：DCdetector: Dual Attention Contrastive Representation Learning for Time Series Anomaly Detection

1. 研究背景

1.1 发展现状

时间序列异常检测在金融市场、经济学、地球科学、制造业、医疗保健等广泛的研究领域和应用中具有重要意义。异常的存在可以表明新的或意想不到的事件，如生产故障、系统缺陷和心悸等，因此特别值得关注。时间序列数据中模式的庞大规模和复杂性促使研究人员开发专门的深度学习模型来检测异常模式。

在过去的几年中，深度学习在学习复杂时间序列的表达表示方面变得越来越有能力，例如同时具有空间(交互)和时间特征的多维数据。在深度异常检测中，神经网络用于学习特征表示或异常分数，以便检测异常。许多深度异常检测模型已经被开发出来，在不同的实际应用中提供了显著高于传统时间序列异常检测任务的性能。在这篇论文中，作者致力于结合对比学习和多尺度双注意力框架，学习一个能够有效区分异常的特征表示。

1.2 时序异常检测简介

异常检测，也称为离群点或新颖性检测，自 20 世纪 60 年代以来一直是众多应用领域的活跃研究领域。随着计算过程的发展，大数据的收集及其在人工智能中的使用得到了更好的实现，有助于包括异常检测在内的时间序列分析。随着数据可用性的增加和算法效率/计算能力的提高，时间序列分析越来越多地用于通过预测、分类和异常检测来解决商业应用。近年来，因为在许多领域都有广泛的应用，包括城市管理、入侵检测等，时间序列异常检测(TSAD)受到了越来越多的关注。

传统的时序数据异常检测方法大多采用统计方法，虽然这些方法具有较高的效率，但是它们没有考虑多元时序数据的长期时间依赖性。在具有复杂结构的数据中，深度神经网络对时间序列中的时间和空间依赖关系具有强大的建模能力，许多学者利用各种深度架构探索了其在异常检测中的应用。

1.3 对比学习简介

在带有噪声干扰的域间或域内任务中，不变信息起着至关重要的作用。例如，用于学习多模态数据的哈希码的一致性表示[1]，利用了多模态数据中的不变信息来构建信息的潜在空间。模态不变非对称网络[2]被用来保持语义相似性，并通过条件变分信息瓶颈网络来学习模态不变信息。

与这些监督方法不同，对比学习方法在自监督的环境中学习数据的变换不变表示。对比学习的主要思想是在特征空间中缩小相似样本之间的距离，在同一空间中最大化不同样本之间的距离。Mo Co [3]进一步考虑到获取负样本的局限性，构建动态字典查找样本对进行对比。Sim CLR [4]讨论了数据增强在对比学习中的重要作用，并提出了一个可学习的非线性变换模块来改进对比学习。将对比学习应用于时间序列数据，TS-TCC [5]提出了一种自监督框架，使用多种对比方法来学习信息丰富的时间序列表示。TS2Vec [6]提出了一种分层对比的方法来学习时间序列的多粒度上下文信息。在这项工作中，这篇论文开发了多元时序数据的对比自编码器表示学习和异常检测。

2. 方法框架

2.1 输入输出

输入：多元时序数据

输出：时间序列异常点时刻标记

2.2 方法概述

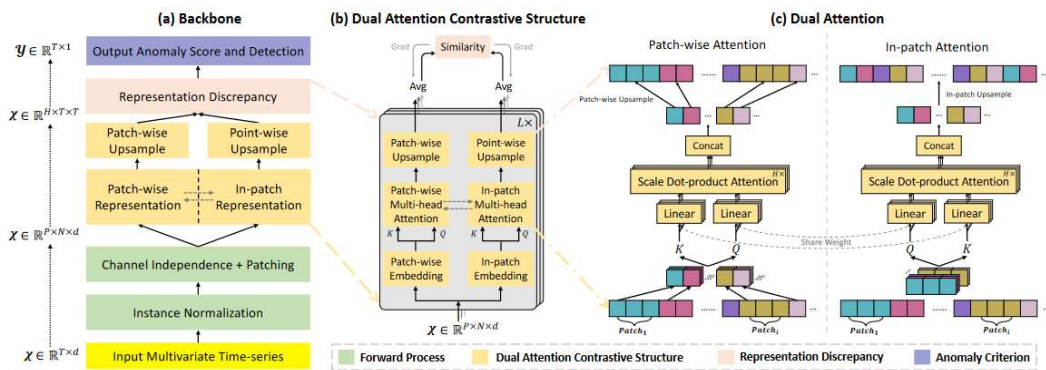


图 2.1 DCDetector 结构图

这篇论文的方法[7]是通过双注意力自监督对比型结构学习输入时间序列的表示分布之间的关联差异来检测异常。其总体架构由 4 个主要组件组成，分别是前向过程模块、双注意力对比结构模块、表征差异模块和异常准则模块。通过实例归一化模块对 Forward Process 模块中的输入多元时间序列进行归一化处理，实例规范化的输入全部来自独立的通道本身，它可以看作是对全局信息的一种巩固和调整，是一种更加稳定的训练处理方式。之后通过不同的视角获取输入时间序列的表示。通过双重注意对比结构，获得两个视角(分片支路和片内支路)的表征。最后形式化基于 KL 散度的损失函数来衡量这两种表示的相似性。

2.3 双重注意对比结构

论文提出了一种具有双重注意力的对比表示学习结构，从不同的视角获取输入时间序列的表示。具体来说，通过 Patching 操作，DC Detector 将 Patch - wise 和 In - Patch 表示作为两个视图。不同于传统的对比学习，其中原始和增强数据被认为是原始数据的两个视图。此外，论文中的方法并不像典型的对比方法那样构建对。相反，它的基本设置类似于只使用正样本的对比方法。对于 Patch-wise 表示，将单个 Patch 视为一个单元，Patch 之间的依赖关系通过多头自注意力网络进行建模。同样，对于 Patch 内表示，通过多头自注意力网络获取同一 Patch 内点的依赖关系。需要注意的是，Patch-wise 注意力网络与 in-Patch 注意力网络共享权重。

为了比较两种表示网络的结果，需要先进行上采样。对于 Patch-wise 分支，由于只有 Patch 之间的依赖关系，所以重复是在 Patch 内部完成的上采样。对于 in-patch 分支，由于只得到补丁点之间的依赖关系，因此从单个 Patch 重复到一组 Patch，得到最终的 in-patch 表示。

3. 实验结果

3.1 论文中的实验结果

Dataset	MSL			SMAP			PSM			SMD		
Metric	P	R	F1	P	R	F1	P	R	F1	P	R	F1
LOF	47.72	85.25	61.18	58.93	56.33	57.60	57.89	90.49	70.61	56.34	39.86	46.68
OCSVM	59.78	86.87	70.82	53.85	59.07	56.34	62.75	80.89	70.67	44.34	76.72	56.19
U-Time	57.20	71.66	63.62	49.71	56.18	52.75	82.85	79.34	81.06	65.95	74.75	70.07
IForest	53.94	86.54	66.45	52.39	59.07	55.53	76.09	92.45	83.48	42.31	73.29	53.64
DAGMM	89.60	63.93	74.62	86.45	56.73	68.51	93.49	70.03	80.08	67.30	49.89	57.30
ITAD	69.44	84.09	76.07	82.42	66.89	73.85	72.80	64.02	68.13	86.22	73.71	79.48
VAR	74.68	81.42	77.90	81.38	53.88	64.83	90.71	83.82	87.13	78.35	70.26	74.08
MMPCACD	81.42	61.31	69.95	88.61	75.84	81.73	76.26	78.35	77.29	71.20	79.28	75.02
CL-MPPCA	73.71	88.54	80.44	86.13	63.16	72.88	56.02	99.93	71.80	82.36	76.07	79.09
TS-CP2	86.45	68.48	76.42	87.65	83.18	85.36	82.67	78.16	80.35	87.42	66.25	75.38
Deep-SVDD	91.92	76.63	83.58	89.93	56.02	69.04	95.41	86.49	90.73	78.54	79.67	79.10
BOCPD	80.32	87.20	83.62	84.65	85.85	85.24	80.22	75.33	77.70	70.9	82.04	76.07
LSTM-VAE	85.49	79.94	82.62	92.20	67.75	78.10	73.62	89.92	80.96	75.76	90.08	82.30
BeatGAN	89.75	85.42	87.53	92.38	55.85	69.61	90.30	93.84	92.04	72.90	84.09	78.10
LSTM	85.45	82.50	83.95	89.41	78.13	83.39	76.93	89.64	82.80	78.55	85.28	81.78
OmniAnomaly	89.02	86.37	87.67	92.49	81.99	86.92	88.39	74.46	80.83	83.68	86.82	85.22
InterFusion	81.28	92.70	86.62	89.77	88.52	89.14	83.61	83.45	83.52	87.02	85.43	86.22
THOC	88.45	90.97	89.69	92.06	89.34	90.68	88.14	90.99	89.54	79.76	90.95	84.99
AnomalyTrans	91.92	96.03	93.93	93.59	99.41	96.41	96.94	97.81	97.37	88.47	92.28	90.33
DCdetector	93.69	99.69	96.60	95.63	98.92	97.02	97.14	98.74	97.94	83.59	91.10	87.18

表 3.1 在真实世界多变量数据集上的总体结果

论文首先在四个真实世界的多变量数据集上用 19 个竞争基线评估了 DCdetector，如表 3.1 所示。可以看出，论文的方法在大多数基准数据集的 F1 度量下取得了 SOTA 结果。

Dataset	Method	Acc	F1	Aff-P [31]	Aff-R [31]	R_A_R [49]	R_A_P [49]	V_ROC [49]	V_PR [49]
MSL	AnomalyTrans	98.69	93.93	51.76	95.98	90.04	87.87	88.20	86.26
	DCdetector	99.06	96.60	51.84	97.39	93.17	91.64	93.15	91.66
SMAP	AnomalyTrans	99.05	96.41	51.39	98.68	96.32	94.07	95.52	93.37
	DCdetector	99.21	97.02	51.46	98.64	96.03	94.18	95.19	93.46
PSM	AnomalyTrans	98.68	97.37	55.35	80.28	91.83	93.03	88.71	90.71
	DCdetector	98.95	97.94	54.71	82.93	91.55	92.93	88.41	90.58

表 3.2 多维度量在真实世界多元数据集上的结果。

论文还进行了与 Anomaly Transformer 的多指标对比，如表 3.2 所示，论文方法在大多数指标上表现较好或至少与 Anomaly Transformer 相当。

Dataset	NIPS-TS-GECCO			NIPS-TS-SWAN		
Metric	P	R	F1	P	R	F1
OCSVM*	2.1	34.1	4.0	19.3	0.1	0.1
MatrixProfile	4.6	18.5	7.4	16.7	17.5	17.1
GBRT	17.5	14.0	15.6	44.7	37.5	40.8
LSTM-RNN	34.3	27.5	30.5	52.7	22.1	31.2
Autoregression	39.2	31.4	34.9	42.1	35.4	38.5
OCSVM	18.5	74.3	29.6	47.4	49.8	48.5
IForest*	39.2	31.5	39.0	40.6	42.5	41.6
AutoEncoder	42.4	34.0	37.7	49.7	52.2	50.9
AnomalyTrans	25.7	28.5	27.0	90.7	47.4	62.3
IForest	43.9	35.3	39.1	56.9	59.8	58.3
DCdetector	38.3	59.7	46.6	95.5	59.6	73.4

表 3.3 在 NIPS - TS 数据集上的整体结果

论文还在表 3.3 中的另外两个数据集 NIPSTS - SWAN 和 NIPS - TS - GECCO 上进行了性能评估，这两个数据集比上述四个数据集更具有挑战性，异常类型更多。尽管两个数据集具有最高的(32.6 %)和最低的(1.1 %)异常比例，论文方法仍然能够取得 SOTA 结果，并且完全优于其他方法。同样，在表 3.4 中进行了 DCdetector 和 Anomaly Transformer 的多指标比较，DCdetector 在大多数指标上仍然取得了更好的性能。

Dataset	Method	Acc	P	R	F1	Aff-P	Aff-R	R_A_R	R_A_P	V_ROC	V_PR
NIPS-TS-SWAN	AnomalyTrans	84.57	90.71	47.43	62.29	58.45	9.49	86.42	93.26	84.81	92.00
	DCdetector	85.94	95.48	59.55	73.35	50.48	5.63	88.06	94.71	86.25	93.50
NIPS-TS-GECCO	AnomalyTrans	98.03	25.65	28.48	26.99	49.23	81.20	56.35	22.53	55.45	21.71
	DCdetector	98.56	38.25	59.73	46.63	50.05	88.55	62.95	34.17	62.41	33.67

表 3.4 多维度量在 NIPS - TS 数据集上的结果

3.2 论文实验复现

3.2.1 源码配置

作者已经提供了项目源代码，链接如下：<https://github.com/DAMO-DI-ML/KDD2023-DCdetector>。源代码中有 10 个文件/文件夹：

data_factory: 预处理文件夹/文件。所有数据集预处理代码都在这里。

dataset: dataset 文件夹，存放所有数据集。

main.py: 主 python 文件。可以在里面调整所有的参数。

metrics: 包含 evaluation metrics code 文件夹，其中包括 VUC、affiliation precision/recall pair 和其他常用指标。

model: DCdetector 模型文件夹。

result: 在代码演示中，可以自动将结果和训练处理日志保存在此文件夹中。

scripts: 所有数据集和消融实验脚本。可以复制实验结果，如 get start 所示。

solver.py: 另一个 python 文件。培训、验证和测试过程都在那里。

utils: 用于数据处理和模型构建的其他函数。

img: readme.md 中需要的图像。

requirements.txt: 运行此存储库所需的 Python 包。

(1) 创建 Conda 环境并安装依赖项

项目中包含 environment.txt 文件用于配置 Conda 环境。选择 Python 版本为 3.10, Pytorch 版本为 1.12, Cuda 版本为 11.3。通过以下指令安装 Conda 环境、激活并安装依赖:

```
conda env create -n dc python==3.10
conda activate dc
conda install pytorch==1.12.0 torchvision==0.13.0 torchaudio==0.12.0
cudatoolkit=11.3 -c pytorch
conda install -r requirement.txt
```

(2) 示例程序

作者提供了 13 组实验脚本进行代码复现, 可以通过下面的指令运行训练和预测:

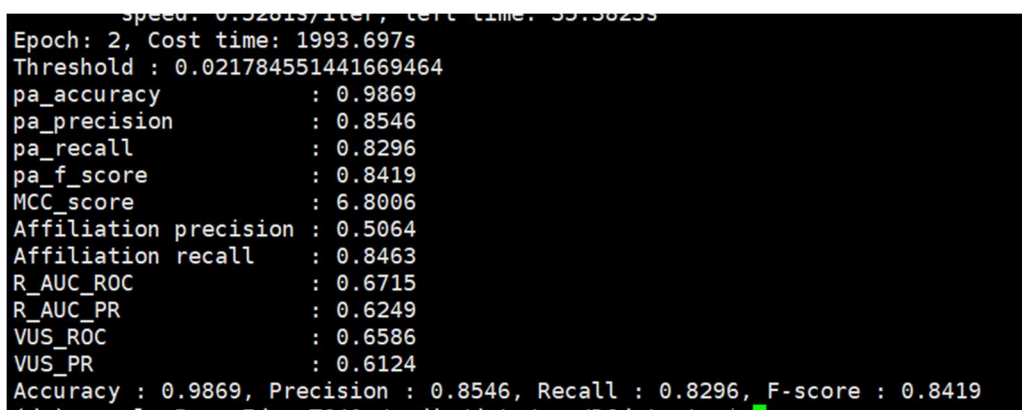
```
bash ./scripts/SMD.sh
bash ./scripts/MSL.sh
bash ./scripts/SMAP.sh
bash ./scripts/PSM.sh
bash ./scripts/SWAT.sh
bash ./scripts/NIPS_TS_Swan.sh
bash ./scripts/NIPS_TS_Water.sh
bash ./scripts/UCR.sh
```

消融实验脚本:

```
bash ./scripts/Ablation_attention_head.sh
bash ./scripts/Ablation_encoder_layer.sh
bash ./scripts/Ablation_Multiscale.sh
bash ./scripts/Ablation_Window_Size.sh
```

(3) 示例程序运行结果

我们选择 SMD 数据集进行代码复现。



```
Epoch: 2, Cost time: 1993.697s
Threshold : 0.021784551441669464
pa_accuracy      : 0.9869
pa_precision     : 0.8546
pa_recall        : 0.8296
pa_f_score       : 0.8419
MCC_score        : 6.8006
Affiliation precision : 0.5064
Affiliation recall  : 0.8463
R_AUC_ROC        : 0.6715
R_AUC_PR         : 0.6249
VUS_ROC          : 0.6586
VUS_PR           : 0.6124
Accuracy : 0.9869, Precision : 0.8546, Recall : 0.8296, F-score : 0.8419
```

图 3.5 运行结果

数据集	MSL			SMAP			PSM			SMD		
度量方法	P	R	F1	P	R	F1	P	R	F1	P	R	F1
DCdetector	93.7	99.7	96.6	95.6	98.9	97.0	97.1	98.7	97.9	83.6	91.1	87.2
ours	94.2	89.5	92.2	94.8	97.3	96.5	98.2	92.5	95.3	85.5	83.0	84.2

表 3.5 对比试验结果

梯度停止		MSL			SMAP			PSM		
Patch-wise	In-patch	P	R	F1	P	R	F1	P	R	F1
×	×	90.1	87.2	88.3	93.3	96.1	95.2	94.6	93.4	95.7
√	×	91.2	70.3	78.4	93.6	92.4	93.6	95.9	97.1	96.4
×	√	91.4	92.6	94.1	93.4	97.6	94.5	96.3	97.3	97.2
√	√	93.2	97.4	96.9	94.2	98.2	97.2	97.1	98.7	97.9

图 3.7 消融实验结果

4 讨论与改进思路

4.1 改进方法概述

数据增强影响着对比学习方法的预测性能，使用更优的数据增强方式能够大大提高模型的预测性能。时序异常检测根据样本是否有异常点来判断该样本是否为正常样本，正常样本的增强样本依旧为正常样本，异常样本的增强样本依旧为异常样本，对样本及其增强样本计算对比损失，使得模型可以学习更优的特征表示。具体操作如下：首先，将原始样本均匀切分为 5 份，随机将这五份进行拼接，形成新的训练样本，作为原始训练样本的正样本，其次，将样本及其增强样本通过 DC detector 获得其对应的特征表示，最后，利用两个特征表示计算对比损失，进而优化整个模型。整个示意图如图 4.1 所示。

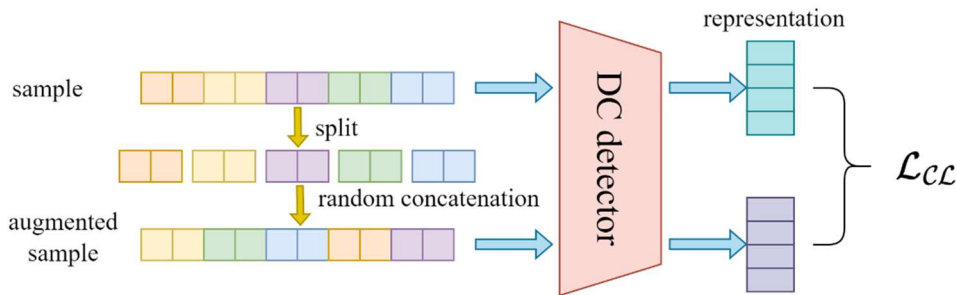


图 4.1 改进示意图

4.2 改进方法实验

表 4.1 展示了改进方法在四个数据集上的实验结果。在不同的数据集上使用不同的度量方法，改进后的方法在大多数的结果上都表现更优，这说明使用数据增强方法可以提高模型的分类准确度。

数据集	MSL			SMAP			PSM			SMD		
度量方法	P	R	F1	P	R	F1	P	R	F1	P	R	F1
<u>DCdetector</u>	93.7	99.7	96.6	95.6	98.9	97.0	97.1	98.7	97.9	83.6	91.1	87.2
ours	94.2	89.5	92.2	94.8	97.3	96.5	98.2	92.5	95.3	85.5	83.0	84.2
<u>New_ours</u>	95.1	92.3	91.4	96.1	97.8	97.5	98.6	94.2	96.7	88.3	85.4	89.8

表 4.1 对比实验

参考文献:

[1] Li L H, Zhang P, Zhang H, et al. Grounded language-image pre-training[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 10965-10975.

[1] Junfeng An, Haoyang Luo, Zheng Zhang, Lei Zhu, Lu. Guangming, Cognitive multi-modal consistent hashing with flexible semantic transformation, Information Processing & Management 59 (1) (2022) 102743.

[2] Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, Maria A Zuluaga, Usad: Unsupervised anomaly detection on multivariate time series, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 3395–3404.

[3] Kaiming He, Haoqi Fan, Wu. Yuxin, Saining Xie, Ross Girshick, Momentum contrast for unsupervised visual representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9729–9738.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, Geoffrey Hinton, A simple framework for contrastive learning of visual representations, in: International Conference on Machine Learning, PMLR, 2020, pp. 1597–1607.

[5] Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee Keong Kwoh, Xiaoli Li, and Cuntai Guan. Time-series representation learning via temporal and contextual contrasting. arXiv:2106.14112, 2021.

[6] Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. arXiv:2106.10466, 2021.

[7] Yang Y, Zhang C, Zhou T, et al. Dcdetector: Dual attention contrastive representation learning for time series anomaly detection[C]//Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2023: 3033-3045.