DCdetector: Dual Attention Contrastive Representation Learning for Time Series Anomaly Detection

Yiyuan Yang[1], Chaoli Zhang[2], Tian Zhou[2], Qingsong Wen[2*], Liang Sun[2]
[1]Department of Computer Science, University of Oxford, [2]DAMO Academy, Alibaba Group
yiyuan.yang@cs.ox.ac.uk, {chaoli.zcl, tian.zt, qingsong.wen, liang.sun}@alibaba-inc.com

Background and Challenges

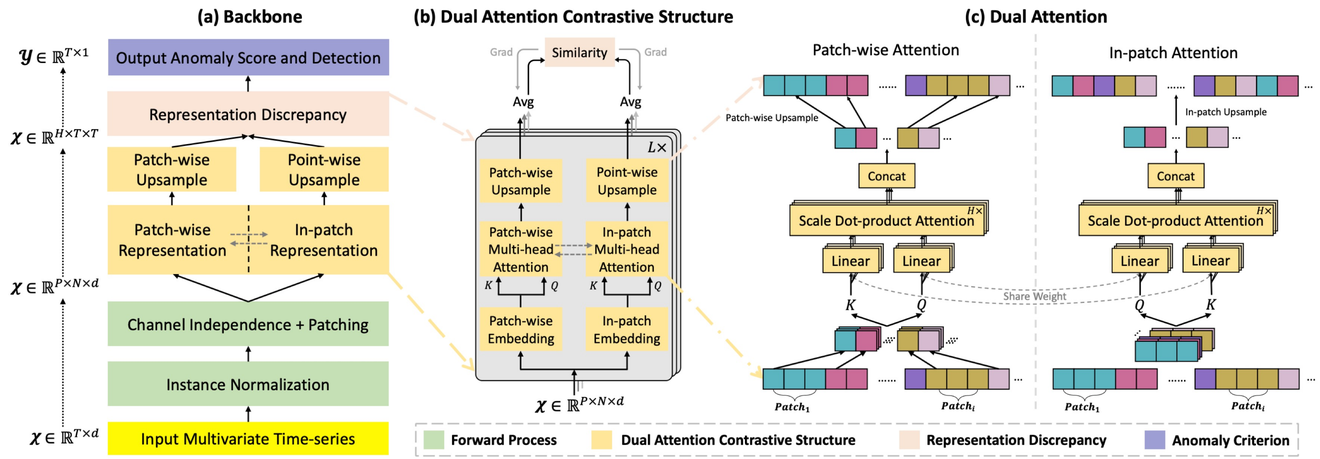
- **Lack of Labeled Data**: Anomalies are usually rare without many labels. Systems are in steady state in major cases.
- **Imbalance**: The number of anomalies is much smaller than the normal one. For example, in the financial system.
- **Noise Interference**: Time-series data may be affected by noise that may mask the true anomaly signal.
- **Complex Patterns**: Typical anomalies are often complex (wind turbines operate in different modes and conditions).
- **Multi-dimensional Features**: Models should consider temporal, multidimensional and non-stationary features.
- **Explanatory and Interpretable**: In some application scenarios, explanatory and interpretable results for anomaly detection are needed to better understand why an anomaly was flagged and to be able to take action accordingly.

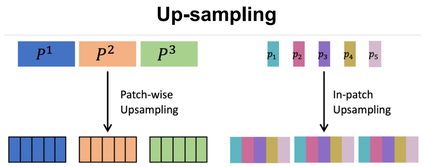
Method: DCdetector's Framework

An intuition: Normal points are closely related to other sample points, while abnormal points are discrete from other sample points. By constructing different representations (Patch-wise and In-patch) between the sample points, if the similarity of the different representations is high, it means that they are normal points.

Two Time-series Representations:

- **Patch-wise representation** $\mathcal{N}$: Attention scores between different patches.
- **In-patch representation** $\mathcal{P}$: Attention scores of internal patch.





Similarity Metric and Loss Function

$$\mathcal{D}(\mathcal{P}, \mathcal{N}) = KL(\mathcal{P}||\mathcal{N})$$

$$\mathcal{L}\{\mathcal{P}, \mathcal{N}; \mathcal{X}\} = \frac{1}{2}\mathcal{D}(\mathcal{P}, \text{Stopgrad}(\mathcal{N})) + \frac{1}{2}\mathcal{D}(\mathcal{N}, \text{Stopgrad}(\mathcal{P}))$$

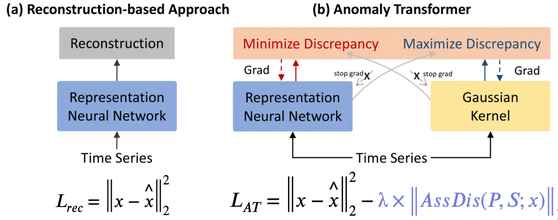
Anomaly Score and Criterion

$$\text{AnomalyScore}(\mathcal{X}) = \frac{1}{2}\mathcal{D}(\mathcal{P}, \mathcal{N}) + \frac{1}{2}\mathcal{D}(\mathcal{N}, \mathcal{P}).$$

$$\mathcal{Y}_i = \begin{cases} 1\text{: anomaly} & \text{AnomalyScore}(\mathcal{X}_i) \geq \delta \\ 0\text{: normal} & \text{AnomalyScore}(\mathcal{X}_i) < \delta \end{cases}$$

Related Works

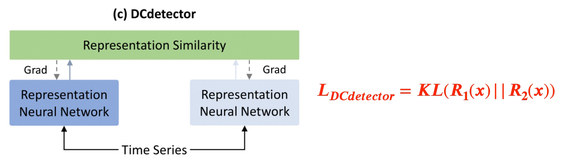
Reconstructed-based problem: The raw time series has a mixture of normalities and anomalies with noise. So it is difficult to train a high quality encoder for reconstruction based models.



$$L_{rec} = \left\| x - \hat{x} \right\|_2^2 \qquad L_{AT} = \left\| x - \hat{x} \right\|_2^2 - \lambda \times \left\| AssDis(P, S; x) \right\|_1$$

Highlight:

DCdetector Conducts Time Series Anomaly Detection without Reconstruction.



$$L_{DCdetector} = KL(R_1(x)||R_2(x))$$

Experiment and Main Results

Table 1: Overall results on real-world multivariate datasets. Performance ranked from lowest to highest. The *P*, *R* and *F1* are the precision, recall and F1-score. All results are in %, the best ones are in Bold, and the second ones are underlined.

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

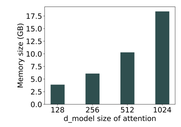
Table 2: Multi-metrics results on real-world multivariate datasets. Aff-P and Aff-R are the precision and recall of affiliation metric [31], respectively. R_A_R and R_A_P are Range-AUC-ROC and Range-AUC-PR [49], which denote two scores based on label transformation under ROC curve and PR curve, respectively. V_ROC and V_RR are volumes under the surfaces created based on ROC curve and PR curve [49], respectively. All results are in %, and the best ones are in Bold.

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

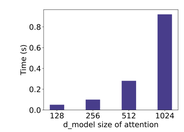
- Datasets & Baselines: **6+1** benchmarks, **26** baselines
- Evaluation Criteria: **10** metrics (F1, Affiliation, VUS)
- Performance on Parameter Sensitivity
- Time-Cost and Memory Used
- Visual Analysis

Table 3: Overall results on NIPS-TS datasets. Performance ranked from lowest to highest. All results are in %, the best ones are in bold, and the second ones are underlined.

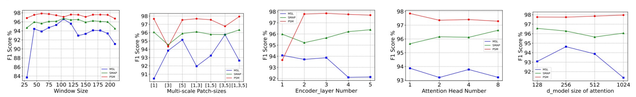
Dataset	NIPS-TS-GECCO			NIPS-TS-SWAN		
Metric	P	R	F1	P	R	F1
OCSVM*	2.1	34.1	4.0	19.3	0.1	0.1
MatrixProfile	4.6	18.5	7.4	16.7	17.5	17.1
GBRT	17.5	14.0	15.6	44.7	37.5	40.8
LSTM-RNN	34.3	27.5	30.5	52.7	22.1	31.2
Autoregression	39.2	31.4	34.9	42.1	35.4	38.5
OCSVM	18.5	**74.3**	29.6	47.4	49.8	48.5
IForest*	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**



(a) Memory used



(b) Time cost



(a) Window size (b) Multi-scale size (c) Encoder layer number (d) Attention head number (e) d_{model} of attention

Figure 6: Parameter sensitivity studies of main hyper-parameters in DCdetector.

Conclusion and Highlight

➢ **Architecture**:
- A contrastive learning-based dual-branch attention structure
- Channel independence patching is proposed to enhance local semantic information
- Multi-scale structure in the attention module can reduce information loss during patching

➢ **Optimization:**
- An effective and robust loss function is designed based on the similarity of two branches
- Model is trained purely contrastively without reconstruction loss, reducing distractions from anomalies

➢ **Performance:**
- DCdetector achieves SOTA performance in 7 benchmarks with 10 metrics, compared with 26 baselines

https://github.com/DAMO-DI-ML/KDD2023-DCdetector