Information Processing and Management 60 (2023) 103383

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Adaptive Multivariate Time-Series Anomaly Detection

Jianming Lv [a,b,*], Yaquan Wang [a], Shengjing Chen [c]

[a] School of Computer Science and Technology, South China University of Technology, Guangzhou, Guangdong, 510006, China
[b] Key Laboratory of Big Data and Intelligent Robot (South China University of Technology), Ministry of Education, Guangzhou, Guangdong, 510006, China
[c] Guangzhou Forsafe Digital Technology Co., Ltd, Guangzhou, Guangdong, 510006, China

ARTICLE INFO

Keywords:
Time-series modeling
Anomaly detection
Domain adaptation
Temporal covariate shirt

ABSTRACT

Existing Multivariate time-series anomaly detection methods aim to calculate the anomaly scores of observed sequences and learn a threshold to judge whether the input data is abnormal. However, they neglected the temporal covariate shift problem, which leads to the learned thresholds cannot be generalized in the test set, resulting in suboptimal detection performance in practical cases. We propose the Adaptive Multivariate Time-series Anomaly Detection framework in this paper, namely DATECT, to address the above challenging problems. Specifically, to enhance the robustness of anomaly measurement, DATECT adopts the dilated convolution based AutoEncoder to integrate both prediction errors and reconstruction errors into the output anomaly scores. Meanwhile, a novel Adaptive Window Normalization method is put forth to reduce the diversity of the distribution of anomaly scores in the test set, hence effectively improving the generalization capability of the detection model. Finally, to further reduce the side-effect of domain-specific dynamic noise, DATECT utilizes Non-parametric Scan Statistics to select the subsets of significantly abnormal signals and highlight the anomaly segments. Experiments on five datasets show that our method can significantly alleviate the performance drop caused by the temporal covariate shift problem, outperforms the baseline in terms of detection performance and generalization, averagely improving the F1-score by 8.66% and the F1*-score (upper bound) by 1.18%.

1. Introduction

The devices in the real world, including servers, spacecraft, and car engines, are outfitted with numerous sensors that can collect multivariate time series of status signals. Automatic anomaly detection on time series can discover anomalies in time and raise alarms to operators to prevent irretrievable loss. With the development of large-scale devices, the increasing number of sensors brings more challenges to effective anomaly detection.

Multivariate time-series anomaly detection (MTAD) methods almost work in an unsupervised mode, because of the scarcity of labeled anomalies, e.g. the distance-based methods (Chaovalitwongse et al., 2007), clustering-based methods (Kiss et al., 2014), and classification-based methods (Liu et al., 2014; Ma & Perkins, 2003). More recently, some deep learning methods measured anomalies based on the prediction errors and reconstruction errors of the model, which showed great advantages in processing high-dimensional time series. In particular, Hundman et al. (2018) leveraged LSTM to detect spacecraft anomalies based on prediction errors. Li et al. (2019) and Zhou et al. (2019) applied GAN to strengthen the prediction model's robustness. Meanwhile, Chen et al. (2022) utilized graph neural networks to model the relationship between various sensors and calculate the prediction errors. Park et al. (2018)

* Corresponding author at: School of Computer Science and Technology, South China University of Technology, Guangzhou, Guangdong, 510006, China.
E-mail addresses: jmlv@scut.edu.cn (J. Lv), csyaquanwang@mail.scut.edu.cn (Y. Wang), shengjing_chen@for-safe.cn (S. Chen).

https://doi.org/10.1016/j.ipm.2023.103383
Received 28 November 2022; Received in revised form 2 April 2023; Accepted 10 April 2023
Available online 17 May 2023

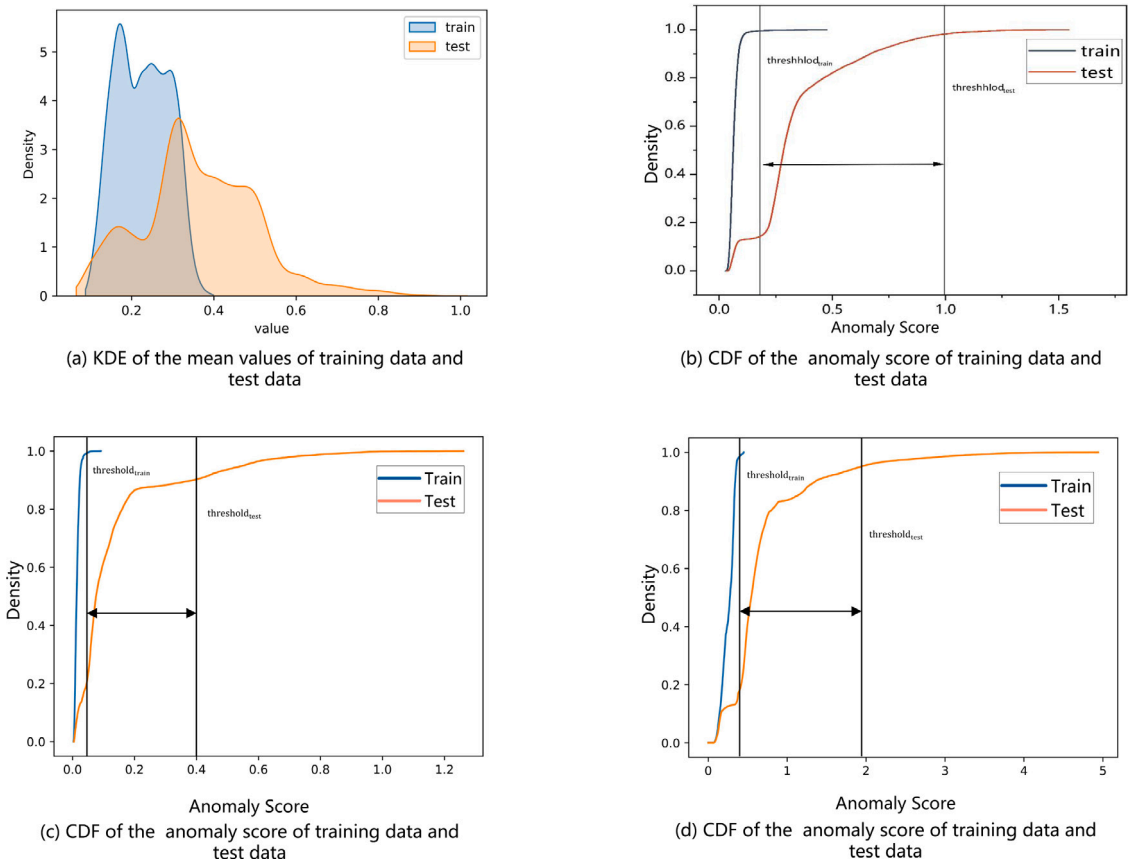


Fig. 1. (a) Kernel density estimation(KDE) of the mean values on the training set and test set of the SMD dataset. (b)~(d) Cumulative distribution(CDF) of the anomaly scores on the training set and test set of the SMD dataset. The anomaly scores of (b) are calculated as the prediction error obtained by MTAD-GAT (Zhao et al., 2020). The anomaly scores of (c) are calculated as the reconstruction error obtained by USAD (Audibert et al., 2020). The anomaly scores of (d) are calculated as the prediction error obtained by GDN (Deng & Hooi, 2021). The optimal threshold of the training set is achieved by POT (Su et al., 2019) method, and the optimal threshold of the test set is achieved by grid search.

and Su et al. (2019) adopted random latent variables to measure anomalies through reconstruction errors. Zhao et al. (2020) further combined both prediction errors and reconstruction errors to score anomalies. Xu et al. (2022) combined reconstruction error and serial correlation to compute anomaly scores. Deng and Hooi (2021) obtained anomaly scores through the learned graph structure and prediction error.

Most of the above methods need to learn a threshold to make the decision about anomalies, when the anomaly score exceeds the threshold, an anomaly is indicated. However, dynamic time series from the real world often exhibit non-stationary property (Quinonero-Candela et al., 2008), and the margin distribution of input data is usually changed over time, causing the temporal covariate shift problem (Du et al., 2021). Even in the same dataset, the distribution of anomaly scores is usually totally different across the training set and test set. Fig. 1 shows the diversity of distributions of the training set and test set in the SMD dataset (Su et al., 2019). Readers can refer to Appendix for a more comprehensive analysis of all datasets in the following experiments. Due to the temporal covariate shift, a large gap can be observed between the distributions on the training set and test set, which means applying the learned thresholds on the training set may not obtain desired results. Table 1 extends the test on three real-world time-series datasets by using state-of-the-art methods, demonstrating the performance drop of transferring thresholds. This limits the detection models' practical application.

To address the aforementioned challenges, we propose the DATECT framework, a novel *Adaptive Multivariate Time-series Anomaly Detection* method. In particular, to enhance the robustness of the anomaly measurement, we adopt the Dilated Convolutional AutoEncoder to integrate both prediction errors and reconstruction errors into the output anomaly scores. Meanwhile, DATECT utilizes a novel *Adaptive Window Normalization* method to reduce the diversity of anomaly score distribution in the test set, so as to enhance the detection model's generalization capacity. To further highlight the anomaly segments and reduce the side-effect of dynamic noise, we utilize the Non-parametric Scan Statistics to select significant anomaly signals. An overview of the paper's main contribution is provided as follows.

Table 1
Detection performance of state-of-the-art methods (MTAD-GAT (Zhao et al., 2020), USAD (Audibert et al., 2020), GDN (Deng & Hooi, 2021)) on three public datasets (SMD (Su et al., 2019), SWaT (Goh et al., 2016), WADI (Ahmed et al., 2017)), F1* represents the results obtained by the optimal threshold of grid search on the test set and F1 represents the result obtained by the optimal threshold on the training set.

Dataset	Methods	F1*	F1	Decline
SMD	MTAD-GAT	86.22	78.41	↓7.81
	USAD	94.63	83.02	↓11.61
	GDN	87.75	79.01	↓8.74
SWaT	MTAD-GAT	84.24	62.18	↓22.06
	USAD	84.60	38.83	↓45.76
	GDN	93.89	77.09	↓16.79
WADI	MTAD-GAT	47.84	15.76	↓32.07
	USAD	42.96	15.90	↓27.07
	GDN	85.52	64.61	↓20.91

- To solve the problem that the training set threshold cannot be generalized to the test set due to the temporal covariate shift, we propose the DATECT framework, a novel adaptive multivariate time-series anomaly detection method. DATECT utilizes a novel Adaptive Window Normalization method to reduce the diversity of the distribution of anomaly scores in the test set, hence effectively improving the generalization capability of the transferred detection model. Meanwhile, DATECT adopts the Dilated Convolutional AutoEncoder to combine both prediction errors and reconstruction errors, so as to increase the robustness of anomaly measurement.
- We adopt the *Non-parametric Scan Statistics* method to further highlight the anomaly segments and reduce the side-effect of dynamic noise by selecting the most important abnormal signals self-adaptively, which also brings better interpretability of detection results.
- The superior performance of DATECT is demonstrated through extensive experiments on five real-world datasets, which significantly alleviate the performance drop caused by the temporal covariate shift problem, and achieve better detection performance and generalization than baseline methods.

2. Related work

2.1. Time series modeling

The capacity to extract representative features from time series is crucial for MTAD. Recently, deep learning-based techniques have shown to be quite useful for modeling high-dimensional time-series data. Specifically, the RNN versions like LSTM (Hochreiter & Schmidhuber, 1997) and GRU (Cho et al., 2014) were able to learn long-term relationships in time series by tackling the vanishing gradient issue. Nguyen et al. (2021) and Liu et al. (2022) modeled temporal data by building an LSTM AutoEncoder. Lin et al. (2020) combined LSTM and variational AutoEncoder to strengthen the robustness of modeling. Besides the tremendous RNN-based models, some recently proposed CNN-based methods (Deep et al., 2020) also showed excellent performance on short-term time series. Ren et al. (2019) utilized spectral analysis and CNN to extract temporal features. Shen et al. (2020) used dilated recurrent convolutional networks to model multi-scale temporal information. Specifically, Bai et al. (2018) applied the causal convolution to achieve larger receptive fields of historical data. Xingjian et al. (2015) further suggested a hybrid model of RNN and CNN, which replaced the dot product in LSTM with convolution operations in order to concurrently evaluate spatial and temporal information. Yin et al. (2022) achieved enhanced temporal modeling by combining CNN and LSTM, which also used multi-level sliding windows to extract features of different scales. Moreover, the attention mechanism was also widely used in time-series data modeling, e.g. Meng et al. (2019) utilized the transformer architecture to capture long-term dependencies through the attention mechanism. Xu et al. (2022) could model longer sequence relationships through transformer and sequence correlation. To enhance the effectiveness of identifying causes, Zhao et al. (2020) applied the attention mechanism in the graph neural network. In addition, graphs are also used to model the relationship between multivariate time series. Deng and Hooi (2021) modeled the correlation between sequences through graphs and extracted features by graph convolution. Li and Jung (2021) used dynamic graphs to model the changing relationship of time series. Chen et al. (2022) further combined transformer and graph convolution to simultaneously obtain temporal features and associations.

2.2. Multivariate time-series anomaly detection

Existing MTAD techniques fall into three categories: *forecasting-based*, *reconstruction-based*, and *dissimilarity-based* methods. Specifically, the forecasting-based methods, e.g. Chen et al. (2022), Deng and Hooi (2021), Ding et al. (2018, 2019) and Hundman et al. (2018), which make a prediction based on historical data and measure the anomaly score as the prediction error. Another main kind of method is reconstruction-based, it learns normal patterns by reconstructing the original input, and measures the anomaly score by reconstruction error. The AutoEncoder framework is popularly used in this kind of technique, e.g. Audibert et al. (2020), Hsieh et al. (2019), Park et al. (2018) and Su et al. (2019). Choi et al. (2020), Li et al. (2019), Tuli et al. (2022) and Zhou et al.

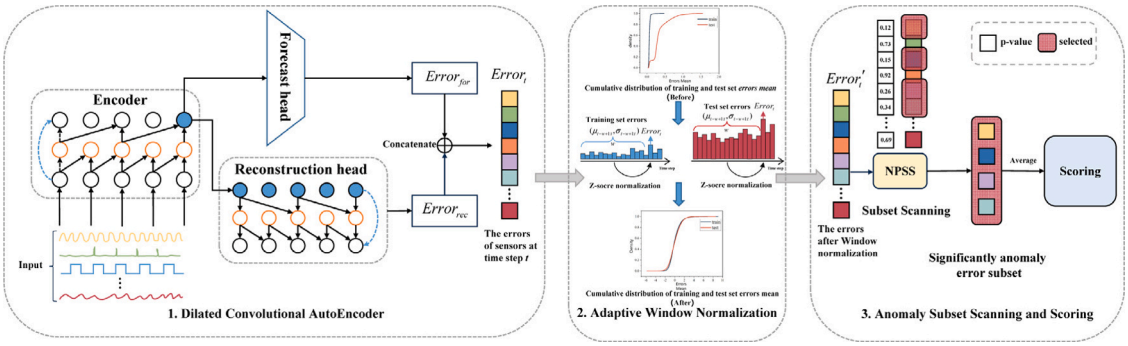


Fig. 2. The overview of DATECT framework, which contains three main parts: (1) AE-Based Forecasting and Reconstructing to predict and reconstruct the value at time t and get the errors $Error_t$. (2) Adaptive Window Normalization to normalize the error at time t by the historical errors with window size w and get the normalized errors $Error_t'$. (3) Subset Scanning and Scoring to select the significant anomaly subset and obtain the anomaly score.

(2019) further integrates the GAN to discriminate the anomaly and normal data. The Dissimilarity-based methods, e.g. He and Zhao (2019), Liu et al. (2019) and Shen et al. (2020) go another way to measure the distance between observed samples and normal ones, where the commonly used distance metrics include Euclidean distance and cosine similarity.

2.3. Domain adaption

According to the notion of domains in the (Farahani et al., 2021), two domains may be different if one of the input space, output space, or joint probability density function varies across domains. Domain adaptation(DA) is a subfield of machine learning that aims to adapt the model trained on the source domain to the target domain well by adjusting the differences between domains. Commonly used methods mainly include instance-based or feature-based methods to align distributions between domains, e.g. Gopalan et al. (2011), Gretton et al. (2009), Jhuo et al. (2012) and Pan et al. (2010). In recent years, deep domain adaptation techniques have been widely used, such as the discrepancy-based Deep Adaption Network (Long et al., 2015) learning cross-domain features. Deep Transfer Network (Zhang et al., 2015) is proposed for matching marginal and conditional distributions simultaneously. Another class of deep domain adaptation methods aligns inter-domain discrepancy via AutoEncoder to minimize reconstruction error and learn cross-domain invariant features, e.g. Bottou (2012), Chen et al. (2012) and Vincent et al. (2010). The Adversarial-based domain adaptation method minimizes cross-domain differences through adversarial learning, generates samples similar to the target domain in the source domain data through the generator, and uses the discriminator to distinguish whether it is a real sample or a generated sample in the target domain, so as to obtain domain-invariant feature, e.g. Ganin and Lempitsky (2015), Pei et al. (2018) and Tzeng et al. (2015).

The problem we want to solve can be viewed as a generalized DA. Due to the existence of temporal covariate shift, the distribution of training data and test data is different, resulting in suboptimal performance of the detection model. But compared to DA, our problem setting and challenges are different. First, the test domain is visible in DA, but the test data in our problem setting is not visible. Secondly, due to the existence of temporal covariate shift, training data and test data are collected in different time periods, resulting in different data distributions. Finally, most DA methods are aimed at classification tasks in images. For time series data, sequence continuity and sequence context are more important.

3. Research objectives

As we stated in Section 1, the main research objective of this paper is to address the detection performance drop due to the temporal covariate shift problem. Existing MTAD methods calculate the anomaly scores of observed sequences and learn a threshold to judge whether the input data is abnormal. However, they neglected the temporal covariate shift problem, which leads to the learned thresholds cannot be generalized in the test set, resulting in suboptimal detection performance in practical cases. As shown in Table 1, we tested three baseline methods(MTAD-GAT Zhao et al., 2020, USAD Audibert et al., 2020, GDN Deng & Hooi, 2021) on three public datasets (SMD Su et al., 2019, SWaT Goh et al., 2016, WADI Ahmed et al., 2017), and the results show that the temporal covariate shift problem affects the performance of the methods in practice use. To this end, a novel adaptive window normalization method is proposed to reduce the diversity of the distribution of anomaly scores, which effectively improves the generalization capability of the transferred detection model. Meanwhile, the Dilated Convolutional AutoEncoder is adopted to combine both prediction errors and reconstruction errors, so as to increase the robustness of anomaly measurement. Additionally, Non-parametric Scan Statistics is utilized to further highlight the anomaly segments and reduce the side-effect of dynamic noise, which brings better interpretability and generalization.

The superior performance of DATECT is demonstrated through extensive experiments on five real-world datasets in Table 3. In Section 5.3 we conducted ablation experiments for each component in the framework to verify the effectiveness of each independent component, and at the same time verified the functions of each component through case study and theoretical analysis. In addition,

we verify the sensitivity and generality of the framework through experiments with different hyperparameter settings in Section 5.4. The results show that our method can significantly alleviate the performance drop caused by the temporal covariate shift problem, and outperforms the baseline in terms of detection performance and generalization, averagely improving the F1-score by 8.66% and the F1*-score (upper bound) by 1.18%.

4. Method

4.1. Problem definition

Detecting whether multi-dimensional time series data is abnormal at a certain time series segment is the goal of MTAD. The input of this problem is defined as the sequence $x = \{x_1, x_2, \dots, x_N\}$, where N is the maximum length of the time steps, and each observation $x_t \in R^k$ is a k-dimensional time series data at the time step $t\,(t \le N)$. The output is the binary decision $y_t \in \{0, 1\}$ of each time step $t\,(t \le N)$ to indicate whether there is an anomaly (1 for anomaly and 0 for normal). The detection model's objective is to determine y_t given the observed historical data in the sliding window of length T, which is denoted as $X_t = [x_{t-T}, x_{t-T+1}, \dots, x_{t-1}] \in R^{k\times T}$. Following the typical unsupervised learning setting in practical use, no label information can be used during training, as labeled anomalies are typically uncommon and difficult to spot.

4.2. Model overview

Fig. 2 shows the overview of our proposed *Adaptive Multivariate Time-series Anomaly Detection* framework, namely DATECT. DATECT is composed of three main components. Firstly, the **Dilated Convolutional AutoEncoder** is designed to capture the deep temporal features from multi-dimensional time series for generating the prediction error and reconstruction error in each time step. Secondly, the **Adaptive Window Normalization** mechanism is adopted to normalize the error sequences, so as to reduce shifts in anomaly score distributions due to temporal covariate shift. Finally, in the **Subset Scanning and Scoring** stage, the most important subsets of error signals are selected to calculate the final anomaly score. The sections that follow will go into detail about each component of DATECT.

4.3. Dilated convolutional AutoEncoder

Forecasting and reconstruction are frequently used ways to measure the anomaly of time series. The abnormal time series, which tend to have different distribution compared with normal data, usually lead to larger forecasting errors (Ding et al., 2019) or reconstruction errors (Su et al., 2019). To enhance the robustness of anomaly measurement, we propose the Dilated Convolutional AutoEncoder framework to combine both of the forecasting and reconstruction errors as shown in Fig. 2. In particular, the encoder consists of multi-layer stacked dilated convolution module (Bai et al., 2018) for capturing the deep temporal features. The encoded feature vector is then fed into the forecasting head and reconstruction head in parallel, where we predict the expected values of each time step and reconstruct the original sequence using historical data.

Dilated Convolutional Encoder Inspired by the convolution modules in Bai et al. (2018), the encoder in Fig. 2 consists of a stack of dilated causal convolution modules with some residual connections to capture multi-scale temporal features. Formally, the input of the model is $X_t = [x_{t-T}, x_{t-T+1}, \dots, x_{t-1}] \in R^{k\times T}$ at the time step t, which is the historical data within the sliding window of length T. Based on the convolutional filters $f_j : \{0, \dots, s-1\} \longrightarrow R^k (j \le D)$, where s is the filter size and D indicates the number of filters, the dilated causal convolution module is defined as follows:

$$F_j(x_t) = \sum_{i=0}^{s-1} f_j(i) \cdot x_{t-d*i} \tag{1}$$

$$z_t = ReLU(F(x_t) + g(x_t)) \tag{2}$$

where Eq. (1) means applying the j_{th} dilated causal convolutional on x_t, and d is the dilation factor. To capture the long-term dependency, d is increased exponentially from the bottom to the top of the stacked layers (i.e. $d = 2^l (0 \le l < L)$, where L represents the total number of network layers.). $F(x_t)$ indicates the concatenation of the collection of $F_j(x_t)(j \le D)$. In addition, the residual block is integrated into the encoder as Eq. (2), where g is 1-D convolution operator for dimension mapping. z_t indicates the temporal feature at time t. The feature vector of the historical sequence X_t is defined as $Z_t = [z_{t-T}, z_{t-T+1}, \dots, z_{t-1}] \in R^{D\times T}$.

Prediction And Reconstruction A multi-head structure is deployed to perform prediction and reconstruction by using the feature vector Z_t output by the encoder. In particular, for the prediction head, predictions of time step t are generated by applying stacked fully-connected layers, which is denoted as $\hat{x}_t$. The reconstruction head adopts multi-layer stacked inverse dilated convolutional layers, and uses a sample-and-hold strategy to repeat the last-step feature of the input for T times. The dilate rates are decreased exponentially to enable the reconstruction from coarse-grained to fine-grained as shown in Fig. 2. The output of the reconstruction head is denoted as $\overline{X}_t = [\overline{x}_{t-T}, \overline{x}_{t-T+1}, \dots, \overline{x}_{t-1}] \in R^{k\times T}$. The loss function can be formulated as:

$$Loss_{pre} = \left\| x_t - \hat{x}_t \right\|_2 \tag{3}$$

$$Loss_{recon} = \frac{1}{T} \sum_{i=1}^{T} \left\| x_{t-i} - \overline{x}_{t-i} \right\|_2 \tag{4}$$

$$Loss_{total} = (1-\alpha) Loss_{recon} + \alpha Loss_{pre} \tag{5}$$

where $Loss_{pre}$ is the forecasting loss, and $Loss_{recon}$ is the reconstruction loss. α is a constant to adjust the balance of these two kinds of loss, and is set to 0.5 by default.

When applying the trained Dilated Convolutional AutoEncoder on the testing data, we can achieve both the forecasting error and reconstruction error on each time step t. Specifically, the forecasting error is defined as $\varepsilon_t = |\hat{x}_t - x_t|$, and the reconstruction error is $r_t = |\overline{x}_t - x_t|$. A larger error means the observed data is more likely to be abnormal, so the anomaly score at time t can be easily defined as:

$$score_t = \|\varepsilon_t\|_1 + \|r_t\|_1 \tag{6}$$

When the $score_t$ exceeds the threshold learned from the training set, the corresponding data point is considered an anomaly.

4.4. Adaptive window normalization

According to our observation shown in Fig. 1, the diversity of anomaly score distribution between the training set and test set usually makes it very hard to transfer the optimal threshold to the unseen test set. Table 1 further shows the significant performance drop due to this temporal covariate shift problem.

Based on the above analysis, we propose a simple but powerful mechanism, namely *Adaptive Window Normalization*, to reduce the temporal covariate shift problem. For each forecasting error ε_t, within the sliding window of length w, its historical error sequence is described as:

$$E_t = [\varepsilon_{t-w}, \varepsilon_{t-w+1}, \dots, \varepsilon_{t-1}] \in R^{w \times k} \tag{7}$$

We apply the statistics of E_t to normalize ε_t as follows:

$$\hat{\varepsilon}_{t,i} = \frac{\varepsilon_{t,i} - \mu(E_{t,i})}{\sigma(E_{t,i})} (i \le k) \tag{8}$$

where $\mu(E_{t,i})$ and $\sigma(E_{t,i})$ are the expectation and variance of $E_{t,i}$ respectively, k is the dimension of the time series.

In a similar way, we can normalize the reconstruction error r_t according to its historical sequence to achieve the normalized result $\hat{r}_t$. Finally, the reconstruction error and forecasting error are concatenated in a $2k$-dimensional error vector:

$$e_t = \hat{\varepsilon}_t \oplus \hat{r}_t \tag{9}$$

where $\oplus$ indicates the concatenation operator.

The above normalization operation is deployed on the training set and test set to greatly reduce the diversity of anomaly score distributions, which facilitates the adaptive threshold selection for anomaly detection.

4.5. Anomaly subset scanning and scoring

Most of the previous error-based methods directly sum up all errors as the anomaly score like Eq. (6). However, due to the domain-specific dynamic noise of time series, this simple strategy may lead to a low SNR, and thus increase the false positive rate. To highlight the effective composition and reduce the side-effect of noise, we adopt the Non-parametric Scan Statistics (NPSS) method (Chen & Neill, 2014; McFowland et al., 2013; McFowland III et al., 2018) to achieve the anomaly score using the most important anomaly signals subset.

The key principle of NPSS is to select the subset with the greatest variation from the background distribution. Due to the scarcity of anomalies, we treat the errors on the training set as background set $D_{train} = \{e_i | i \le M\}$, where e_i is the i_{th} error vector sampled from the training set and is calculated as Eq. (9). The training set has a size of M. Given the testing data point at time t with the error vector e_t, the p-value of $e_{t,j} (j \le 2k)$ is defined as:

$$p_{t,j} = \frac{\sum_{e_i \in D_{train}} I(e_{i,j} \ge e_{t,j}) + 1}{M + 1}, \tag{10}$$

which indicates the proportion of the errors exceeding $e_{t,j}$ in the background set. Here $I(.)$ is the indicator function. A Smaller p-value indicates a more significant variation from the background distribution, and means more likely to be an anomaly. Given any 2k-dimensional error e_t of the test data at the time t, the problem addressed here is to select out the subset of dimensions, which have significant deviations. To measure the total deviation of a selected subset S, the NPSS score is defined as follows according to McFowland et al. (2013):

$$F(S) = \max_{\alpha} F_{\alpha}(S) = \max_{\alpha} \phi_{BJ}(\alpha, N_{\alpha}(S), N(S)) \tag{11}$$

$$\phi_{BJ}(\alpha, N_{\alpha}, N) = N * KL(\frac{N_{\alpha}}{N}, \alpha) \tag{12}$$

where $N(S)$ is the size of S, $N_{\alpha}(S)$ is the number of p-values smaller than α, α is the significance level which indicates how uncommon the p-value is. Eq. (12) is the BJ test statistic (Berk & Jones, 1979), where KL is the Kullback–Leibler divergence $KL(x, y) = x log \frac{x}{y} + (1-x) log \frac{1-x}{1-y}$.

Based on the NPSS score, the complete procedure to select the anomaly subset is detailed in Algorithm 1. Because scanning all possible combinations of the subset can lead to exponential time complexity, inspired by Cintas et al. (2020), we exploit LTSS (Neill, 2012) strategy by sorting the p-values to improve the efficiency.

Algorithm 1 NPSS over train error set and test error set

Input: D_{train}: Background error set. e_t: Test error at time t. α_{max}: Significance threshold.
Output: Anomaly subset S_t.

for $j = 1$ *to* $2k$ **do**
  $p_{t,j} = \frac{\sum_{e_i \in D_{train}} I(e_{i,j} >= e_{t,j}) + 1}{M+1}$
end
$p_t^s = \{p_{t,j} | p_{t,j} < \alpha_{max}, j \leq k\}$
$p_t^* = Sort(p_t^s)$
for $n = 1$ *to* $2k$ **do**
  $S_{(n)} = \{p_{t,1}^*, p_{t,2}^*, ..., p_{t,n}^*\}$
  $\alpha_n = \max\{S_{(n)}\}$
  $F(S_{(n)}) = \phi_{BJ}(\alpha_n, n, n)$
end
$n^* = argmax_n(F(S_{(n)}))$;
$S_t = S_{(n^*)}$;
return S_t;

After achieving the anomaly subset S_t by Algorithm 1, the anomaly score of Eq. (6) is rewritten as follows to sum up the selected errors:

$$\overline{score}_t = \frac{\sum_{e_{t,i} \in S_t} e_{t,i}}{n^*} \tag{13}$$

where the subset has a size of n^*, $e_{t,i}$ is the error in the selected subset S_t. For automatic threshold selection, we apply the Peak Over Threshold(POT) (Su et al., 2019) on the training set to obtain the optimal threshold th^*. The idea of the POT method is to fit the data extreme value distribution (tail distribution) through the generalized Pareto distribution (GPD). We calculate the anomaly scores of the training set $S_{train} = \{S_1, S_2, \ldots, S_M\}$, and then set an initial threshold th via empirical quantile. We use the GPD function as follows:

$$\bar{F}(s) = \mathbb{P}(S_{train} - th > s \mid S_{train} > th) \sim \left(1 + \frac{\gamma x}{\beta}\right)^{-\frac{1}{\gamma}} \tag{14}$$

where γ and β is the parameter of GPD. $S_{train} - th$ indicates the extreme value of the anomaly score that exceeds the initial threshold th. As in Siffer et al. (2017), we use Maximum Likelihood Estimation to estimate the value of $\hat{\gamma}$ and $\hat{\beta}$, and the optimal threshold th^* is calculated by:

$$th^* \simeq th + \frac{\hat{\beta}}{\hat{\gamma}} \left(\left(\frac{qM}{M_{th}} \right)^{-\hat{\gamma}} - 1 \right) \tag{15}$$

where q is an expected probability that $S_{train} > th$, M is the size of the training set, and M_{th} is the size of anomaly scores in the training set greater than the initial threshold. In the POT method, the empirical quantile and q need to be adjusted empirically. Finally, a data point with a $\overline{score}_t$ exceeds the threshold th^* will be considered an anomaly.

4.6. Offline training and online detection

Our proposed framework is divided into two phases, the offline training phase, and the online detection phase. In particular, the offline training phase optimizes the Dilated Convolutional AutoEncoder through the prediction task and the reconstruction task jointly, and trains the prediction head and the reconstruction head simultaneously. In the online detection phase, the trained model is used to obtain the prediction errors and reconstruction errors. The anomaly score is obtained for anomaly detection through the Adaptive Window Normalization module and the Anomaly Subset Scanning and Scoring module.

Offline Training Specifically, the training set is divided into sliding windows of length T as described in Section 4.1. The Dilated Convolutional AutoEncoder, prediction head, and reconstruction head are optimized according to Eq. (5) until convergence. The trained model is then applied to the training set to obtain the prediction errors and reconstruction errors, which are fed into the Adaptive Window Normalization module and the Abnormal Subset Scanning and Scoring module to obtain the anomaly scores of the training set. The optimal threshold th^* is also achieved by using the Peak Over Threshold (Su et al., 2019) method.

Online Detection The test data is divided into sliding windows of length T in the same way as the offline training phase. Following the similar pipeline, we can achieve the anomaly score, which is used to compare with the threshold th^* obtained in the offline training phase. A higher score will be announced as a detected anomaly.

Table 2
Statistics of benchmark datasets.

Dataset	#Train	#Test	#Features	Anomalies
MSL	58,317	73,729	51	10.5%
SMAP	135,183	427,617	25	12.8%
SMD	708,400	708,400	38	4.21%
SWaT	495,00	449,919	51	12.33%
WADI	784,570	172,801	123	5.77%

5. Experiments

5.1. Experimental settings

Datasets The statistics of five widely used multivariate time series datasets are summarized in Table 2.

- **MSL and SMAP** (Hundman et al., 2018) came from anomaly data from spacecraft sensors collected by NASA.
- **SMD** (Su et al., 2019) was collected from the server of the internet company. The detection results are the averaged values from 28 sub-datasets.
- **SWaT** (Goh et al., 2016) was collected from Singapore's Public Utility Board's scaled-down water treatment test-bed.
- **WADI** (Ahmed et al., 2017) came from a reduced city water distribution system.

Baseline models. We compare our method with the following state-of-the-art methods:

- **OCSVM** (Tax & Duin, 2004): OCSVM is based on the classic machine learning method SVM, and learns the decision boundary between normal and abnormal data.
- **IsolationForest** (Liu et al., 2008): IsolationForest calculates the anomaly score by building the isolation tree to divide high-density regions and low-density regions.
- **LOF** (Breunig et al., 2000): LOF is a density-based local outlier detection algorithm.
- **DAGMM** (Zong et al., 2018): DAGMM applies a jointing deep AutoEncoder and Gaussian mixture model.
- **LSTM-VAE** (Park et al., 2018): LSTM-VAE is an LSTM-based variational AutoEncoder that calculates the anomaly score as reconstruction error.
- **MAD-GAN** (Li et al., 2019): MAD-GAN applies a GAN-based model to compute the anomaly score.
- **OmniAnomaly** (Su et al., 2019): OmniAnomaly applies GRU-based model to calculate the reconstruction likelihood as the anomaly score.
- **MTAD-GAT** (Zhao et al., 2020): MTAD is a GAT-based method to model the relationship between sensors.
- **USAD** (Audibert et al., 2020): USAD combines AutoEncoder with an adversarial training phase.
- **GDN** (Deng & Hooi, 2021): GDN applies the graph neural network to learn the relationship between sensors while detecting anomalies.
- **AnomalyTrans** (Xu et al., 2022): AnomalyTrans uses transformer to model time series and combines time series correlation for anomaly detection.

Evaluation Metrics. The detection performance is evaluated using the F1-score: $F1 = \frac{2\times Precision\times Recall}{Precision+Recall}$. In practice, abnormal observations typically happen repeatedly to create contiguous anomaly segments. Following the evaluation strategy in Su et al. (2019), If any time step in the abnormal segment is detected correctly, the detection on the whole segment is treated as correct.

Implement Details. The model is trained up to 50 epochs on the server with 8 NVIDIA 1080Ti graphics cards. We set the Adam optimizer's learning rate to 1×10^{-3}, while the early stopping with patience 5 is applied. The length of the sliding window is set to 100. Correspondingly, in order to make the receptive field reach 100, we stack 7 layers of dilated convolution module with filter size 3. Hidden state dimension D is set to 64, and the balance parameter α in Eq. (5) is set to 0.5. The max-threshold α_{max} in Algorithm 1 is set to 0.99. For the automatic threshold selection method POT (Su et al., 2019) used in DATECT, the empirical quantile and probability q are set to 0.9 and 0.001 respectively.

5.2. Performance analysis

We compare DATECT with state-of-the-art methods as shown in Table 3 to illustrate the superior performance of our method, where the precision, recall, F1, and F1* for all methods are reported. In particular, F1 represents the precision when the optimal thresholds are obtained from the training set. F1* indicates the upper bound of the precision when using the optimal threshold on the test set obtained by the grid search on the whole dataset.

In general, OCSVM (Tax & Duin, 2004), IsolationForest (Liu et al., 2008), LOF (Breunig et al., 2000) and DAGMM (Zong et al., 2018), which are based on traditional machine learning methods, perform relatively poorly. Most of these methods are

Table 3
Performance of our method and other baselines from 5 real-world temporal datasets. The Total column is the average of the results of all datasets. F1 represents the result obtained by the optimal threshold on the training set and the F1* represents the result obtained by the optimal threshold of grid search on the test set, reproduced results are marked as †.

Methods	SMAP				MSL				SMD			
	Prec	Rec	F1	F1*	Prec	Rec	F1	F1*	Prec	Rec	F1	F1*
OCSVM	53.85	59.07	56.34	–	59.78	86.87	70.82	–	44.34	76.72	56.19	–
IsolationForest	52.39	59.07	55.53	–	53.94	86.54	66.45	–	42.31	73.29	53.64	–
LOF	58.93	56.33	57.60	–	47.72	85.25	61.18	–	56.34	39.86	46.68	–
DAGMM	58.54	90.58	71.12	82.06	54.12	99.34	70.07	86.16	59.51	87.82	70.95	72.78
LSTM-VAE	85.51	63.66	72.98	75.73	52.57	95.46	67.80	73.88	79.22	70.75	74.75	79.28
MAD-GAN	80.56	67.60	73.51	81.31	59.23	96.80	73.49	87.47	72.05	87.13	78.88	81.03
OmniAnomaly	74.16	97.76	84.34	85.31	88.67	91.17	89.90	89.90	83.34	94.49	88.57	94.40
MTAD-GAT	91.47	91.23	91.35†	93.67	87.54	94.40	90.84†	91.86	71.16	87.29	78.40†	86.22
USAD	70.12	98.31	81.86†	86.34	85.19	97.86	91.09†	92.72	81.34	84.77	83.02†	94.63
GDN	84.15	96.79	90.03†	90.58	82.92	99.19	90.33†	93.06	70.06	90.57	79.01†	87.75
AnomalyTrans	93.55	95.62	94.57†	96.69	90.77	83.39	86.92†	93.59	72.08	83.63	76.15†	92.33
DATECT	93.87	99.48	**96.59**	**96.95**	92.68	96.93	**94.76**	**95.05**	88.68	95.68	**92.05**	**95.33**
Methods	SWaT				WADI				Total			
	Prec	Rec	F1	F1*	Prec	Rec	F1	F1*	Prec	Rec	F1	F1*
OCSVM	45.39	49.22	47.23	–	–	–	–	–	50.84	67.97	57.64	–
IsolationForest	49.29	44.95	47.02	–	–	–	–	–	49.48	65.96	55.66	–
LOF	72.15	65.43	68.62	–	–	–	–	–	58.78	61.72	58.52	–
DAGMM	15.45	96.21	26.62†	79.71	7.65	99.99	14.21†	20.94	39.05	94.79	50.59	68.33
LSTM-VAE	–	–	–	80.51	–	–	–	37.99	72.43	76.62	–	69.48
MAD-GAN	22.53	90.99	36.12†	83.17	7.94	78.22	14.42†	31.62	48.46	84.15	55.28	72.92
OmniAnomaly	44.17	89.9	59.24†	83.28	9.55	99.99	17.43†	41.74	59.98	94.66	67.90	78.93
MTAD-GAT	49.88	82.54	62.18†	84.24	8.66	87.35	15.76†	47.84	61.74	88.56	67.71	80.77
USAD	24.40	95.01	38.83†	84.6	8.69	93.27	15.90†	42.96	53.95	93.84	62.14	80.25
GDN	63.94	97.05	77.09†	93.89†	76.28	56.03	64.61†	85.52†	75.47	87.93	80.22	90.16
AnomalyTrans	82.88	98.08	89.84†	94.07	68.95	93.56	77.44†	86.93†	81.67	90.86	84.98	92.72
DATECT	88.13	99.99	**93.69**	**94.64**	77.42	93.27	**84.61**	**87.12**	88.16	97.07	**92.34**	**93.82**

transferred from the univariate time-series modeling while neglecting the relationship between multiple variables in historical observations. LSTM-VAE (Park et al., 2018) and OmniAnomaly (Su et al., 2019) use recurrent neural networks (RNN) to detect anomalies by calculating reconstruction probability. Since the effect of reconstruction depends on the modeling of time series data, the information-forgetting problem will occur in the RNN-based methods. The dilated convolution-based method (Bai et al., 2018) outperforms the RNN-based methods on different temporal modeling tasks, while the receptive field of dilated convolution is the entire sequence. MTAD-GAT (Zhao et al., 2020) and GDN (Deng & Hooi, 2021) utilize graph convolutional networks (Zhou et al., 2020) to model the dependency of different signals and calculate the prediction error as anomaly score. All of these graph-based methods depend on the correctness of the graph structure. While the system is lack prior knowledge about the correct dependency of the signals, the network will be overfitting to the wrong signal relationship, thereby degrading the performance. MAD-GAN (Li et al., 2019) and USAD (Audibert et al., 2020) are based on generative adversarial methods and calculate anomaly score based on reconstruction error. AnomalyTrans (Xu et al., 2022) is based on the transformer and uses reconstruction error to calculate the anomaly score.

Table 3 shows a significant performance gap between F1 and F1* in previous works, especially in some challenging datasets such as SMD, SWaT, and WADI. This confirms the serious temporal covariate problem of existing methods. Our methods can achieve much higher scores of both F1 and F1*. The slight performance drop between F1 and F1* of DATECT also shows its excellent adaptation capability.

Furthermore, we performed a visual analysis of each dataset in Appendix and compared the visualization results of each dataset with the experimental results in Table 3. It can be observed from Fig. 9 that the temporal covariate shift in SMAP (Hundman et al., 2018) is not serious, and accordingly, the gap between F1 and F1* of most methods in Table 3 is small. For the other four datasets, the temporal covariate shift is obvious in Figs. 10–13, which causes the significant gap between F1 and F1* of most baseline methods in Table 3.

5.3. Ablation studies

To intuitively explain how our method works, we provide ablation studies and visualization results of four critical designs in DATECT in the following sections: the Adaptive Window Normalization strategy, anomaly subset scanning (NPSS), Dilated Convolutional AutoEncoder, and jointing optimization of prediction errors with reconstruction errors.

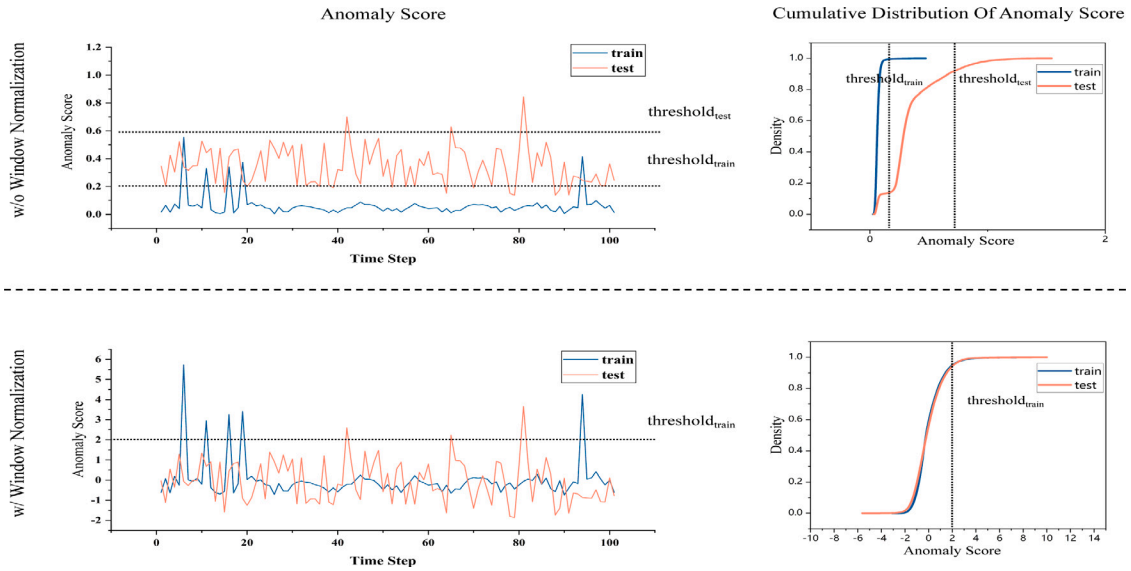


Fig. 3. On the left are the anomaly score in the training set and test set without and with Adaptive Window Normalization (where the test set Anomaly Score produced a mean shift), and on the right is the cumulative distribution of anomaly score without and with Adaptive Window Normalization on SMD dataset.

Table 4
Ablation study on benchmark datasets. W indicates the Adaptive Window Normalization. N indicates NPSS.

Module		MSL		SMAP		SMD		SWaT		WADI		Total	
W	N	F1	F1*	F1	F1*	F1	F1*	F1	F1*	F1	F1*	F1	F1*
		89.69	90.48	81.88	84.67	50.04	86.07	80.70	83.17	16.82	46.80	63.83	78.24
	✓	88.43	88.71	81.20	85.81	53.58	88.30	82.15	83.23	71.43	71.48	75.36	83.51
✓		91.97	94.06	92.04	97.23	90.97	95.72	91.01	93.84	80.72	90.07	89.34	94.18
✓	✓	**94.76**	95.05	**96.59**	96.95	**92.05**	95.33	**93.69**	94.64	**84.61**	87.12	**92.34**	93.82

5.3.1. Effectiveness of adaptive window normalization

Fig. 3 shows the effectiveness of the Adaptive Window Normalization, which can significantly reduce the diversity of anomaly score distribution between the training set and test set. In particular, for visualization purposes, we calculate the average of the errors called anomaly score. The left top of Fig. 3 shows a mean drift of anomaly score distributions, which may lead to poor generalization of the threshold. The right top of Fig. 3 further shows the diversity of the cumulative anomaly score distribution, which confirms the temporal covariate shift. The bottom sub-figures show the results after using the Adaptive Window Normalization, which effectively reduces the diversity of anomaly score distributions and enables the optimized thresholds achieved from the training set to be easily transferred to the test set. Meanwhile, the normalization does not weaken the contrast between adjacent normal and abnormal segments.

Table 4 further shows the performance of the DATECT model equipped with the Adaptive Window Normalization. While removing the normalization module, a significant performance drop can be observed on DATECT, confirming this component's importance.

5.3.2. Effectiveness of NPSS

The ablation study in Table 4 shows that the model combining NPSS with the Adaptive Window Normalization can achieve the best performance. Fig. 4 further illustrates the power of NPSS in selecting important anomaly subsets of signals. The original reconstruction errors on each time step are illustrated on the left heatmap, which shows much dynamic noise. The p-values of the selected signals from subset scanning are shown in the right heatmap, which highlights the most important anomalies and filters out dynamic noise. The time series with smaller average p-values are shown on the rightmost sub-figure and the detected abnormal segments are highlighted, which enables precise locating of anomalies in practical use.

Fig. 5 further explores the effectiveness of the NPSS and window normalization techniques on detecting different types of anomalies. Here we follow the same criteria for classifying anomalies as (Lai et al., 2021). Overall, the anomaly scores of the full DATECT model are more discriminative compared to the baseline models without NPSS and Adaptive Window Normalization. DATECT is able to highlight the anomalies while suppressing the normal points in both contextual and pattern anomalies, thus reducing the number of omissions and false positives. In contrast, the anomaly scores of the baselines are more noisy and jittery, which may lead to the wrong decision of anomaly detection.

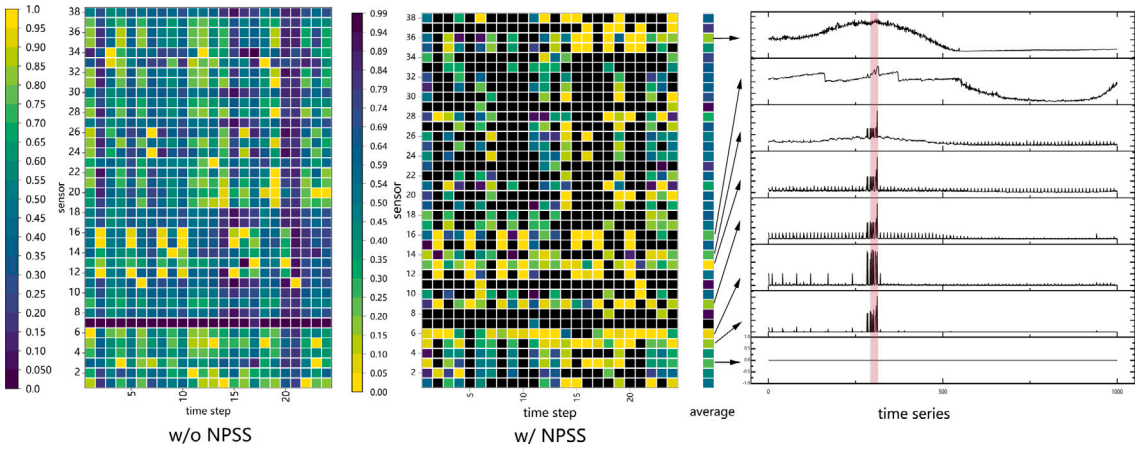


Fig. 4. Case study for NPSS on SMD dataset. The left heatmap shows the errors on each time step, and the right heatmap shows the p-values of the selected subset, where the filtered signals are painted in black. The rightmost sub-figure shows the time series with smaller average p-values, which indicate a higher possibility of anomalies. The vertical axis represents data dimensions, and the numeric value of error/p-value is represented by the color scale in the legend. Each small square represents the error/p-value of a certain sensor at a certain time step. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

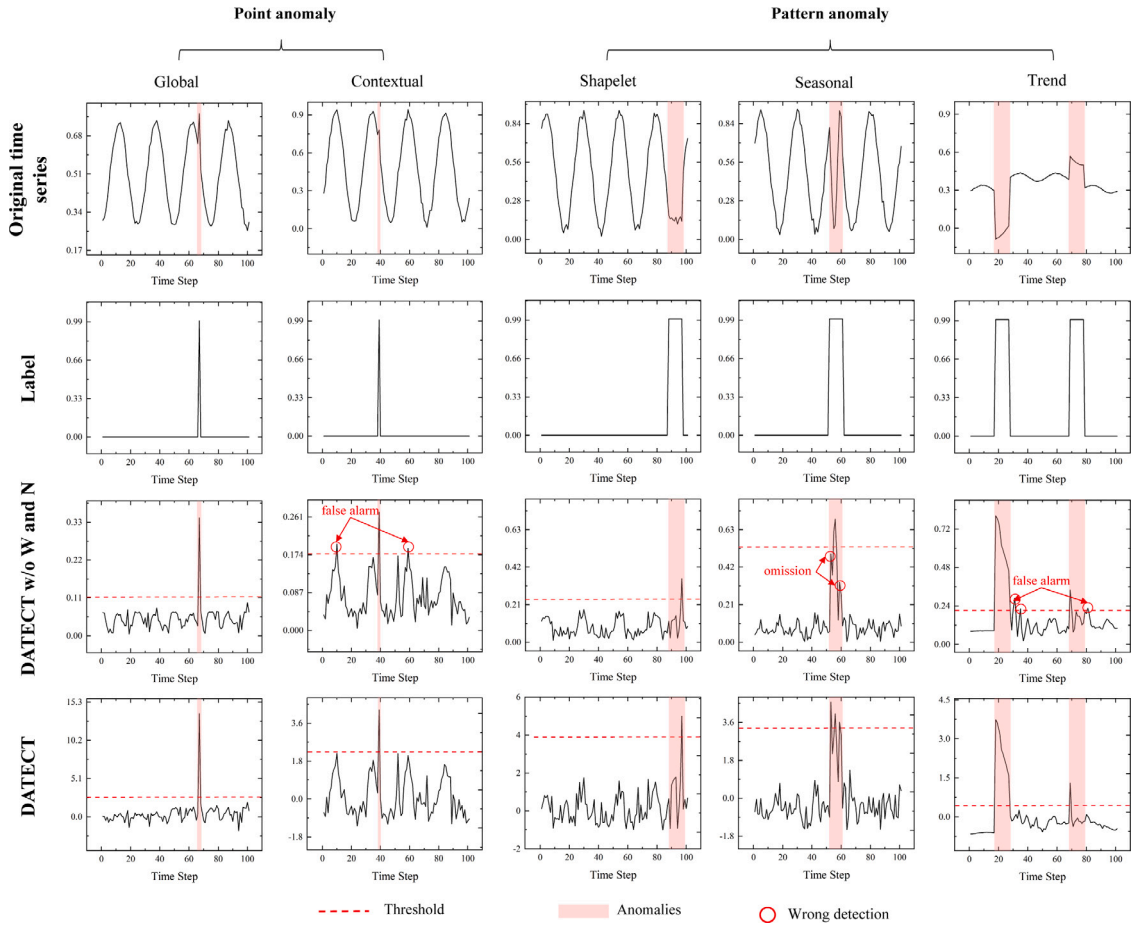


Fig. 5. Visualization of different categories of anomaly. The first row plots the original time series from the TS dataset, and the second row shows the corresponding labels. The third and fourth rows show the original anomaly scores and the anomaly scores obtained using our method. Among the anomalous segments we use red boxes to indicate them and circle the wrong detections in red.

Table 5
Performance of different backbone on five datasets. W indicates the Adaptive Window Normalization. N indicates NPSS.

Backbone	Module		MSL		SMAP		SMD		SWaT		WADI		Total	
	W	N	F1	F1*	F1	F1*	F1	F1*	F1	F1*	F1	F1*	F1	F1*
GAT-based			93.67	95.04	82.53	88.88	48.41	86.22	27.84	84.24	15.77	47.84	53.64	80.44
		✓	82.23	92.2	80.52	82.23	48.01	86.84	28.39	84.89	16.3	48.00	51.09	78.83
	✓		92.93	94.29	92.27	97.84	90.05	95.62	92.14	92.90	78.52	86.14	89.18	93.36
	✓	✓	94.89	96.26	96.61	97.33	85.91	93.13	90.05	93.03	62.07	86.13	85.91	93.18
GAN-based			45.54	78.75	63.47	70.13	47.44	81.03	36.12	60.70	14.42	31.62	41.40	64.45
		✓	45.86	76.23	62.62	66.54	50.92	77.79	87.12	89.99	14.01	27.88	52.11	67.69
	✓		72.97	79.28	86.01	91.86	79.60	93.75	91.50	92.66	82.74	86.03	82.56	88.72
	✓	✓	66.83	80.39	86.32	88.33	74.54	87.57	91.13	92.12	60.67	81.90	75.90	86.06
Transformer-based			73.49	87.21	73.32	79.60	46.64	87.75	27.11	81.30	14.87	44.05	47.09	75.98
		✓	74.86	80.59	79.80	87.64	50.63	83.61	27.23	82.91	34.02	67.49	53.31	80.45
	✓		69.07	92.86	92.86	97.92	93.23	95.66	92.08	92.24	65.99	88.50	82.65	93.44
	✓	✓	94.70	94.13	98.18	98.40	91.45	96.18	89.73	90.90	84.50	91.17	91.71	94.16
DilatedConv-based(Ours)			89.69	90.48	81.88	84.67	50.04	86.07	80.70	83.17	16.82	46.80	63.83	78.24
		✓	88.43	88.71	81.20	85.81	53.58	88.30	82.15	83.23	71.43	71.48	75.36	83.51
	✓		91.97	94.06	92.04	97.23	90.97	95.72	91.01	93.84	80.72	90.07	89.34	94.18
	✓	✓	94.76	95.05	96.59	96.95	92.05	95.33	93.69	94.64	84.61	87.12	**92.34**	93.82

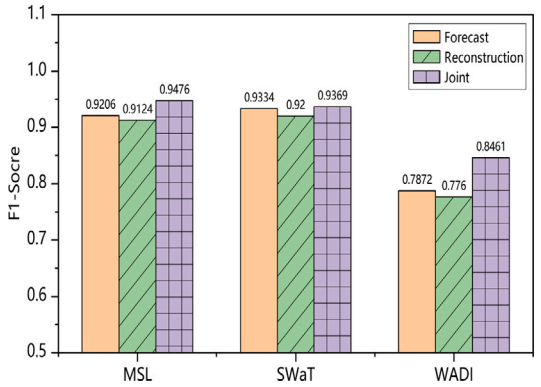


Fig. 6. Results under different optimization strategies.

5.3.3. *Effectiveness of dilated convolutional AutoEncoder*

To explore the effectiveness of Dilated Convolutional AutoEncoder, which serves as the feature extractor for time-series processing, we replace the backbone network with the graph convolution-based network (Zhao et al., 2020), GAN-based network (Geiger et al., 2020), and Transformer-based network (Zhou et al., 2021) respectively. As shown in Table 5, DATECT outperforms other variation models in most cases. This benefits from the superior performance of Dilated Convolution AutoEncoder in the following two aspects. Firstly, the multi-level dilated convolutional network is easier to capture multi-scale long-term dependency in time-series compared with traditional convolutional network. Secondly, its network structure is much more lightweight than other complex backbones, which increases the adaptation ability of the model and decreases the gap between F1 and F1*.

5.3.4. *Effectiveness of multi-task optimization*

As shown in Fig. 2, the multi-task optimization is applied in DATECT, which corresponds to the forecasting and reconstruction heads in the Dilated Convolutional AutoEncoder. In this section, we perform an ablation study to verify the effectiveness of the combination of these two tasks, and the results are shown in Fig. 6. The performance of the model with only forecasting head is slightly better than the reconstruction-based model. However, they are both worse than the multi-task-based model. In fact, the reconstruction-based model is better at capturing the overall distribution of time series, such as periodicity, while the forecasting-based model is more susceptible to sudden data disturbances and changes. Their combination can fully utilize their benefits to achieve better anomaly detection performance.

5.4. *Parameter analysis*

As a crucial AutoEncoder framework parameter, the historical time window T affects the performance of forecasting and reconstruction on time series. A smaller historical time window will be more sensitive to short-term prediction and reconstruction,

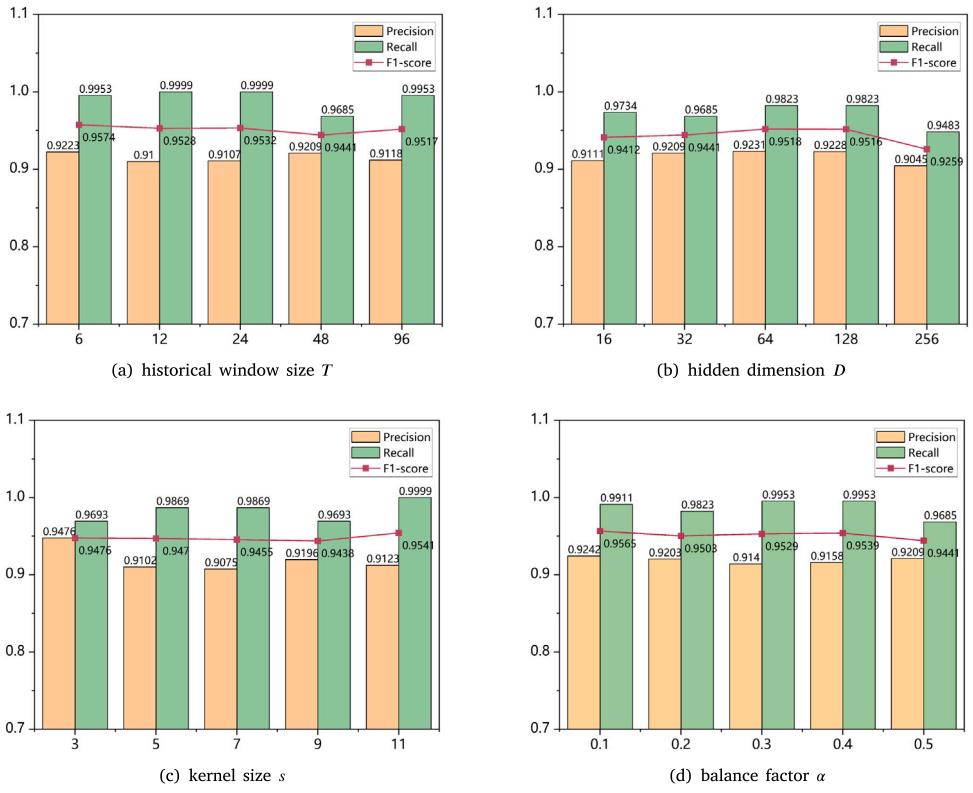


Fig. 7. Parameter analysis for (a) historical window size T. (b) hidden dimension D. (c) kernel size s. (d) balance factor α. All experiments are performed on the MSL dataset.

and yields quick detection of dynamic changes. On the contrary, a larger historical time window will pay more attention to periodic and seasonal changes, and prefer long-term anomalies. Fig. 7(a) shows the results of DATECT under different window sizes. It can be observed that the performance is not sensitive to the changing of T, which shows the stability of DATECT.

Fig. 7(b) shows the impact of the dimension D of hidden space, which is related to the size of the feature space. When D is relatively smaller, the decoder will be more difficult to decode the hidden vector well, which contains relatively less information. When we gradually increase D to the other extreme, the performance drops significantly. This is because an overlarge D may greatly increase the complexity of the model and cause over-fitting on the training data. Fig. 7(b) shows that the optimized D is 64, which is also proved effective on all datasets.

Receptive field size is influenced by the size of the convolution kernel. The performance of DATECT with different sizes of convolution kernels is shown in Fig. 7(c). Performance only slightly improves with increasing convolution kernel size s. A larger convolution kernel usually means higher computational complexity of the model, so in practice, we use the convolution kernels with smaller size (3 by default) and stack multiple convolutional layers to achieve a large receptive field.

According to Eq. (5), the prediction loss and the reconstruction loss are balanced by the constant α. When it is set to 0 or 1, the training task is dominated by prediction or reconstruction. Fig. 7(d) shows how the performance changes when adjusting α from 0.1 to 0.5. The performance of DATECT under different settings is not much different, indicating that DATECT is not sensitive to α.

We also analyze how the size of the normalization window w (Eq. (7)) affects the performance in Fig. 8. It can be observed that when the window size is small, the detection performance is weak. This is because the short-term statistical characteristics are highly variable, and the distribution of error values is not steady, resulting in the degradation of detection performance. When the window size gradually increases, the performance is improved and tends to stabilize. The extreme case may be taking all historical errors for standardization. However, it is impossible to adopt a too-large window size. In fact, this will cause long-term cold boot problems and result in high delay. Meanwhile, because the anomalies are usually continuous and of different sizes, the optimal window size in various data sets may be different. In practice, we generally choose a medium-sized window size (100 by default), which is a trade-off between performance and cold boot.

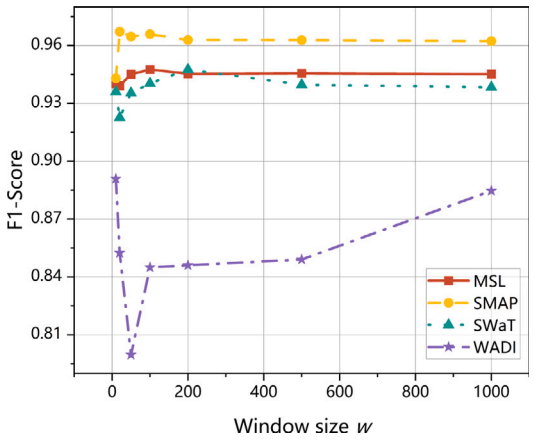


Fig. 8. Effect of different normalization window sizes on detection performance.

6. Conclusion

In this work, we propose the DATECT framework to address the temporal covariate shift problem in time-series anomaly detection. *DATECT* is based on the dilated causal convolution network which uses an Adaptive Window Normalization method to reduce the anomaly score distribution between the training set and the test set. It also utilizes the Non-parametric Scan Statistics to select significant anomaly subsets to highlight the anomaly segments and reduce the side-effect of dynamic noise. The superior performance of DATECT is demonstrated through extensive experiments on five real-world datasets. Strengthening the method's interpretability and confirming its practical effects in more complex scenarios will be the main goals of future work.

CRediT authorship contribution statement

Jianming Lv: Conceptualization, Methodology, Investigation, Formal analysis, Writing – original draft. **Yaquan Wang:** Conceptualization, Methodology, Investigation, Data curation, Writing – original draft. **Shengjing Chen:** Resources, Data curation, Writing – review & editing.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the Special Fund Project of Marine Economy Development in Guangdong Province([2021]35) and the Science and Technology Program of Guangzhou,China (2023B03J1388).

Appendix. Multivariate time series dataset temporal covariate shift

To illustrate more detail about the temporal covariate shift problem in multivariate time series, we performed a visual analysis of five real-world datasets in Figs. 10–13. The introduction of the datasets can be found in Section 5.1. We use the MTAD-GAT (Zhao et al., 2020), USAD (Audibert et al., 2020), GDN (Deng & Hooi, 2021) to calculate the anomaly scores. The anomaly scores of MTAD-GAT are calculated based on the prediction error, while that of USAD and GDN are based on the reconstruction error and prediction error respectively. For ease of visualization, we average the multivariate data at each time step to obtain the mean values. The subfigure (a) of each dataset is the Kernel Density Estimation(KDE) of the time series. In particular, Fig. 9(a) shows that the distribution of training data and test data in the SMAP (Hundman et al., 2018) dataset is basically similar, while in the other four datasets (Figs. 10–13) distinct temporal covariate shift can be observed. Meanwhile, subgraph (b)~(d) of each dataset show the cumulative distribution function (CDF) of the anomaly scores for MTAD-GAT, USAD, and GDN, respectively. It can be observed that the different degree of the anomaly scores between training data and test data is related to the degree of the temporal covariate shift. Higher covariate shift may lead to a larger diversity of the distribution of anomaly scores.

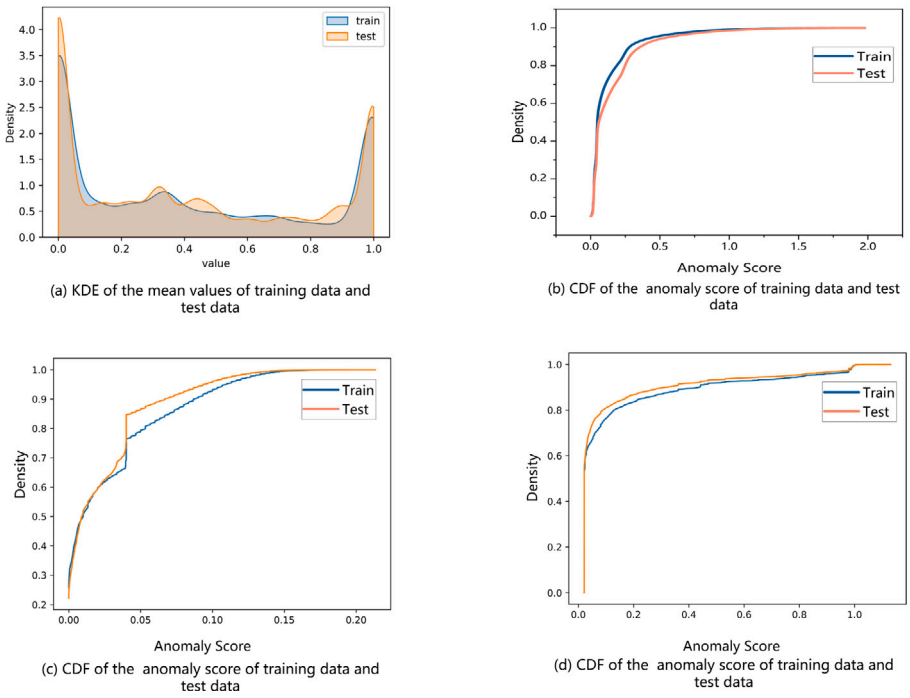


Fig. 9. Visual analysis of SMAP (Hundman et al., 2018) dataset. Subgraph (a) is the kernel density estimate of the training data and test data. Subgraphs (b)~(d) are the cumulative distribution function (CDF) of the anomaly scores for MTAD-GAT, USAD, and GDN, respectively.

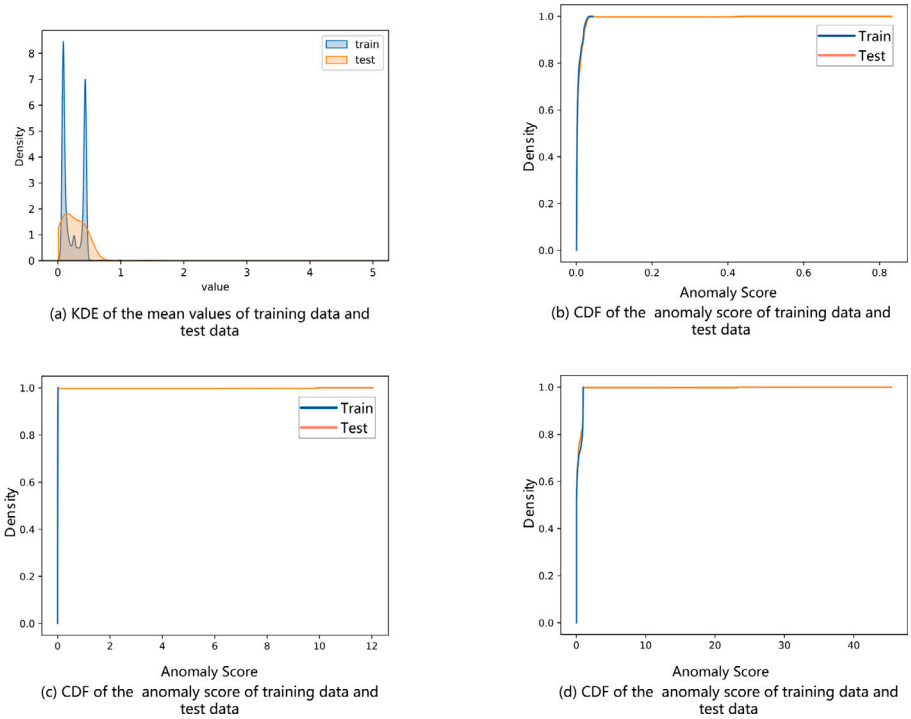


Fig. 10. Visual analysis of MSL (Hundman et al., 2018) dataset. Subgraph (a) is the kernel density estimate of the training data and test data. Subgraphs (b)~(d) are the cumulative distribution function (CDF) of the anomaly scores for MTAD-GAT, USAD, and GDN, respectively.

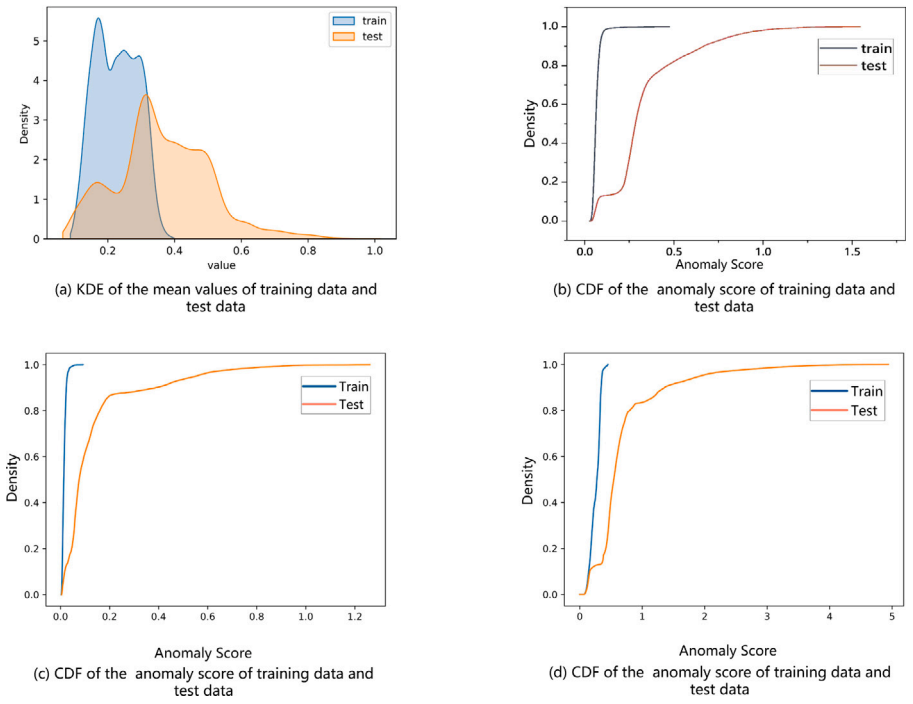


Fig. 11. Visual analysis of SMD (Su et al., 2019) dataset. Subgraph (a) is the kernel density estimate of the training data and test data. Subgraphs (b)~(d) are the cumulative distribution function (CDF) of the anomaly scores for MTAD-GAT, USAD, and GDN, respectively.

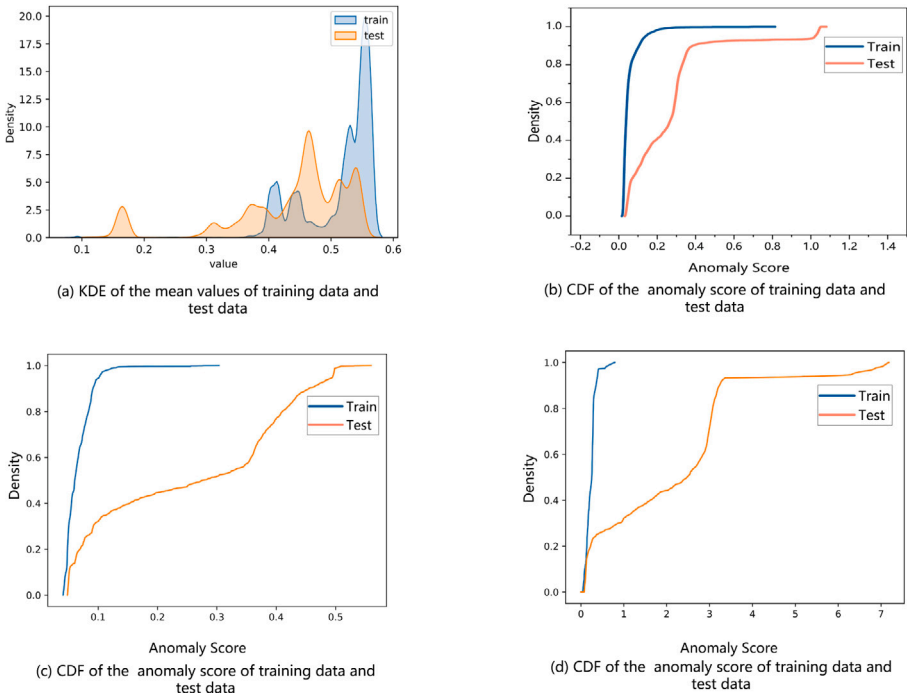
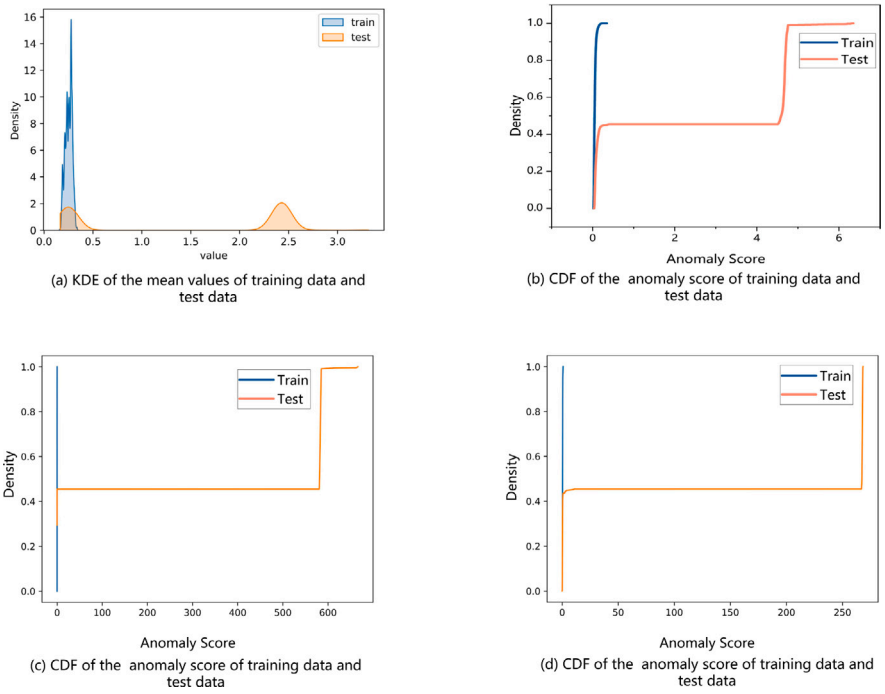


Fig. 12. Visual analysis of SWAT (Goh et al., 2016) dataset. Subgraph (a) is the kernel density estimate of the training data and test data. Subgraphs (b)~(d) are the cumulative distribution function (CDF) of the anomaly scores for MTAD-GAT, USAD, and GDN, respectively.



(a) KDE of the mean values of training data and test data

(b) CDF of the anomaly score of training data and test data

(c) CDF of the anomaly score of training data and test data

(d) CDF of the anomaly score of training data and test data

Fig. 13. Visual analysis of WADI (Ahmed et al., 2017) dataset. Subgraph (a) is the kernel density estimate of the training data and test data. Subgraphs (b)~(d) are the cumulative distribution function (CDF) of the anomaly scores for MTAD-GAT, USAD, and GDN, respectively.

References

Ahmed, C. M., Palleti, V. R., & Mathur, A. P. (2017). WADI: A Water Distribution Testbed for Research in the Design of Secure Cyber Physical Systems. In *Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks, CySWATER@CPSWeek 2017, Pittsburgh, Pennsylvania, USA, April 21, 2017* (pp. 25–28).

Audibert, J., Michiardi, P., Guyard, F., Marti, S., & Zuluaga, M. A. (2020). USAD: unsupervised anomaly detection on multivariate time series. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020* (pp. 3395–3404). ACM.

Bai, S., Kolter, J. Z., & Koltun, V. (2018). An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. arXiv preprint arXiv:1803.01271.

Berk, R. H., & Jones, D. H. (1979). Goodness-of-fit Test Statistics that Dominate the Kolmogorov Statistics. *Zeitschrift für Wahrscheinlichkeitstheorie Und Verwandte Gebiete*, *47*(1), 47–59.

Bottou, L. (2012). Stochastic Gradient Descent Tricks. In *Neural networks: Tricks of the trade: Second edition* (pp. 421–436). Springer.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). LOF: Identifying Density-Based Local Outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA* (pp. 93–104).

Chaovalitwongse, W. A., Fan, Y.-J., & Sachdeo, R. C. (2007). On the Time Series K-Nearest Neighbor Classification of Abnormal Brain Activity. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *37*(6), 1005–1016.

Chen, Z., Chen, D., Zhang, X., Yuan, Z., & Cheng, X. (2022). Learning Graph Structures With Transformer for Multivariate Time-Series Anomaly Detection in IoT. *IEEE Internet of Things Journal*, *9*(12), 9179–9189.

Chen, F., & Neill, D. B. (2014). Non-Parametric Scan Statistics for Event Detection and Forecasting in Heterogeneous Social Media Graphs. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014* (pp. 1166–1175).

Chen, M., Xu, Z., Weinberger, K., & Sha, F. (2012). Marginalized Denoising AutoEncoders for Domain Adaptation. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. Omnipress.

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL* (pp. 1724–1734). ACL.

Choi, Y., Lim, H., Choi, H., & Kim, I. (2020). GAN-Based Anomaly Detection and Localization of Multivariate Time Series Data for Power Plant. In *2020 IEEE International Conference on Big Data and Smart Computing, BigComp 2020, Busan, Korea (South), February 19-22, 2020* (pp. 71–74). IEEE.

Cintas, C., Speakman, S., Akinwande, V., Ogallo, W., Weldemariam, K., Sridharan, S., & McFowland, E. (2020). Detecting Adversarial Attacks via Subset Scanning of AutoEncoder Activations and Reconstruction Error. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020* (pp. 876–882).

Deep, S., Tian, Y., Lu, J., Zhou, Y., & Zheng, X. (2020). Leveraging Multi-view Learning for Human Anomaly Detection in Industrial Internet of Things. In *2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics), iThings/GreenCom/CPSCom/SmartData/Cybermatics 2020, Rhodes Island, Greece, November 2-6, 2020* (pp. 533–537).

Deng, A., & Hooi, B. (2021). Graph Neural Network-Based Anomaly Detection in Multivariate Time Series. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021* (pp. 4027–4035).

Ding, N., Gao, H., Bu, H., & Ma, H. (2018). RADM: Real-time Anomaly Detection in Multivariate Time Series Based on Bayesian Network. In *2018 IEEE International Conference on Smart Internet of Things (SmartIoT), Xi'an, China, August 17-19, 2018* (pp. 129–134). IEEE.

Ding, N., Ma, H., Gao, H., Ma, Y., & Tan, G. (2019). Real-time Anomaly Detection Based on Long Short-Term Memory and Gaussian Mixture Model. *Computers & Electrical Engineering*, *79*, Article 106458.

Du, Y., Wang, J., Feng, W., Pan, S., Qin, T., Xu, R., & Wang, C. (2021). Adarnn: Adaptive Learning and Forecasting of Time Series. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021* (pp. 402–411).

Farahani, A., Voghoei, S., Rasheed, K., & Arabnia, H. R. (2021). A Brief Review of Domain Adaptation. In *Advances in data science and information engineering: proceedings from ICDATA 2020 and IKE 2020* (pp. 877–894). Springer.

Ganin, Y., & Lempitsky, V. (2015). Unsupervised Domain Adaptation by Backpropagation. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015* (pp. 1180–1189). PMLR.

Geiger, A., Liu, D., Alnegheimish, S., Cuesta-Infante, A., & Veeramachaneni, K. (2020). TadGAN: Time Series Anomaly Detection Using Generative Adversarial Networks. In *2020 IEEE International Conference on Big Data (IEEE BigData 2020), Atlanta, GA, USA, December 10-13, 2020* (pp. 33–43). IEEE.

Goh, J., Adepu, S., Junejo, K. N., & Mathur, A. (2016). A Dataset to Support Research in the Design of Secure Water Treatment Systems. In *Critical Information Infrastructures Security - 11th International Conference, CRITIS 2016, Paris, France, October 10-12, 2016, Revised Selected Papers* (pp. 88–99). Springer.

Gopalan, R., Li, R., & Chellappa, R. (2011). Domain Adaptation for Object Recognition: An Unsupervised Approach. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011* (pp. 999–1006). IEEE.

Gretton, A., Smola, A., Huang, J., Schmittfull, M., Borgwardt, K., & Schölkopf, B. (2009). Covariate Shift by Kernel Mean Matching. *Dataset Shift in Machine Learning*, *3*(4), 5.

He, Y., & Zhao, J. (2019). Temporal Convolutional Networks for Anomaly Detection in Time Series. *1213*, In *Journal of Physics: Conference Series*. (4), IOP Publishing, Article 042050.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-term Memory. *Neural Computation*, *9*(8), 1735–1780.

Hsieh, R.-J., Chou, J., & Ho, C.-H. (2019). Unsupervised Online Anomaly Detection on Multivariate Sensing Time Series Data for Smart Manufacturing. In *12th IEEE Conference on Service-Oriented Computing and Applications, SOCA 2019, Kaohsiung, Taiwan, November 18-21, 2019* (pp. 90–97). IEEE.

Hundman, K., Constantinou, V., Laporte, C., Colwell, I., & Soderstrom, T. (2018). Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018* (pp. 387–395).

Jhuo, I.-H., Liu, D., Lee, D., & Chang, S.-F. (2012). Robust Visual Domain Adaptation with Low-Rank Reconstruction, booktitle=2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012. (pp. 2168–2175). IEEE.

Kiss, I., Genge, B., Haller, P., & Sebestyén, G. (2014). Data Clustering-based Anomaly Detection in Industrial Control Systems. In *2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP)* (pp. 275–281). IEEE.

Lai, K.-H., Zha, D., Xu, J., Zhao, Y., Wang, G., & Hu, X. (2021). Revisiting Time Series Outlier Detection: Definitions and Benchmarks. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Li, D., Chen, D., Jin, B., Shi, L., Goh, J., & Ng, S.-K. (2019). MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks. In *Artificial Neural Networks and Machine Learning - ICANN 2019: Text and Time Series - 28th International Conference on Artificial Neural Networks, Munich, Germany, September 17-19, 2019, Proceedings, Part IV* (pp. 703–716). Springer.

Li, G., & Jung, J. J. (2021). Dynamic Graph Embedding for Outlier Detection on Multiple Meteorological Time Series. *PLoS One*, *16*(2), Article e0247119.

Lin, S., Clark, R., Birke, R., Schönborn, S., Trigoni, N., & Roberts, S. (2020). Anomaly Detection for Time Series Using VAE-LSTM Hybrid Model. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020* (pp. 4322–4326). IEEE.

Liu, T., Pu, Y., Shi, J., Li, Q., & Chen, X. (2014). Towards Misdirected Email Detection for Preventing Information Leakage. In *IEEE Symposium on Computers and Communications, ISCC 2014, Funchal, Madeira, Portugal, June 23-26, 2014* (pp. 1–6).

Liu, P., Sun, X., Han, Y., He, Z., Zhang, W., & Wu, C. (2022). Arrhythmia Classification of LSTM AutoEncoder Based on Time Series Anomaly Detection. *Biomedical Signal Processing and Control*, *71*, Article 103228.

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation Forest. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy* (pp. 413–422). IEEE.

Liu, J., Zhu, H., Liu, Y., Wu, H., Lan, Y., & Zhang, X. (2019). Anomaly detection for time series using temporal convolutional networks and Gaussian Mixture Model. *1187*, In *Journal of Physics: Conference Series*. (4), IOP Publishing, Article 042111.

Long, M., Cao, Y., Wang, J., & Jordan, M. (2015). Learning Transferable Features with Deep Adaptation Networks, booktitle=Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015. (pp. 97–105). PMLR.

Ma, J., & Perkins, S. (2003). Time-Series Novelty Detection Using One-Class Support Vector Machines. In *Proceedings of the international joint conference on neural networks, 2003, Vol. 3* (pp. 1741–1745). IEEE.

McFowland, E., Speakman, S., & Neill, D. B. (2013). Fast Generalized Subset Scan for Anomalous Pattern Detection. *Journal of Machine Learning Research*, *14*(1), 1533–1561.

McFowland III, E., Somanchi, S., & Neill, D. B. (2018). Efficient Discovery of Heterogeneous Treatment Effects in Randomized Experiments via Anomalous Pattern Detection. arXiv preprint arXiv:1803.09159.

Meng, H., Zhang, Y., Li, Y., & Zhao, H. (2019). Spacecraft Anomaly Detection via Transformer Reconstruction Error. In *Proceedings of the International Conference on Aerospace System Science and Engineering 2019* (pp. 351–362). Springer.

Neill, D. B. (2012). Fast Subset Scan for Spatial Pattern Detection. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, *74*(2), 337–360.

Nguyen, H., Tran, K., Thomassey, S., & Hamad, M. (2021). Forecasting and Anomaly Detection Approaches Using LSTM and LSTM AutoEncoder Techniques with the Applications in Supply Chain Management. *International Journal of Information Management*, [ISSN: 0268-4012] *57*, 102282.

Pan, S. J., Tsang, I. W., Kwok, J. T., & Yang, Q. (2010). Domain Adaptation via Transfer Component Analysis. *IEEE Transactions on Neural Networks*, *22*(2), 199–210.

Park, D., Hoshi, Y., & Kemp, C. C. (2018). A Multimodal Anomaly Detector for Robot-Assisted Feeding Using an LSTM-Based Variational AutoEncoder. *IEEE Robotics and Automation Letters*, *3*(3), 1544–1551.

Pei, Z., Cao, Z., Long, M., & Wang, J. (2018). Multi-Adversarial Domain Adaptation. In S. A. McIlraith, & K. Q. Weinberger (Eds.), *Proceedings of the thirty-second AAAI conference on artificial intelligence, (AAAI-18), the 30th innovative applications of artificial intelligence (IAAI-18), and the 8th AAAI symposium on educational advances in artificial intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018* (pp. 3934–3941). AAAI Press.

Quinonero-Candela, J., Sugiyama, M., Schwaighofer, A., & Lawrence, N. D. (2008). *Dataset Shift in Machine Learning*. Mit Press.

Ren, H., Xu, B., Wang, Y., Yi, C., Huang, C., Kou, X., Xing, T., Yang, M., Tong, J., & Zhang, Q. (2019). Time-Series Anomaly Detection Service at Microsoft. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019* (pp. 3009–3017).

Shen, L., Li, Z., & Kwok, J. (2020). Timeseries Anomaly Detection using Temporal Hierarchical One-Class Network. *Advances in Neural Information Processing Systems*, *33*, 13016–13026.

Siffer, A., Fouque, P.-A., Termier, A., & Largouet, C. (2017). Anomaly Detection in Streams with Extreme Value Theory. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017* (pp. 1067–1075).

Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., & Pei, D. (2019). Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019* (pp. 2828–2837).

Tax, D. M., & Duin, R. P. (2004). Support Vector Data Description. *Machine Learning*, *54*(1), 45–66.

Tuli, S., Casale, G., & Jennings, N. R. (2022). TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data. *Proc. VLDB Endow.*, *15*(6), 1201–1214.

Tzeng, E., Hoffman, J., Darrell, T., & Saenko, K. (2015). Simultaneous Deep Transfer Across Domains and Tasks. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015* (pp. 4068–4076).

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A., & Bottou, L. (2010). Stacked Denoising AutoEncoders: Learning Useful Representations in A Deep Network with A Local Denoising Criterion. *Journal of Machine Learning Research*, *11*(12).

Xingjian, S., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., & Woo, W.-c. (2015). Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1* (pp. 802–810).

Xu, J., Wu, H., Wang, J., & Long, M. (2022). Anomaly Transformer: Time Series Anomaly Detection with Association Discrepancy. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

Yin, C., Zhang, S., Wang, J., & Xiong, N. N. (2022). Anomaly Detection Based on Convolutional Recurrent AutoEncoder for IoT Time Series. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *52*(1), 112–122.

Zhang, X., Yu, F. X., Chang, S.-F., & Wang, S. (2015). Deep Transfer Network: Unsupervised Domain Aaptation. arXiv preprint arXiv:1503.00591.

Zhao, H., Wang, Y., Duan, J., Huang, C., Cao, D., Tong, Y., Xu, B., Bai, J., Tong, J., & Zhang, Q. (2020). Multivariate Time-Series Anomaly Detection via Graph Attention Network. In *20th IEEE International Conference on Data Mining, ICDM 2020, Sorrento, Italy, November 17-20, 2020* (pp. 841–850). IEEE.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., & Sun, M. (2020). Graph Neural Networks: A Review of Methods and Applications. *AI Open*, *1*, 57–81.

Zhou, B., Liu, S., Hooi, B., Cheng, X., & Ye, J. (2019). BeatGAN: Anomalous Rhythm Detection using Adversarially Generated Time Series. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019* (pp. 4433–4439).

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021* (pp. 11106–11115).

Zong, B., Song, Q., Min, M. R., Cheng, W., Lumezanu, C., Cho, D., & Chen, H. (2018). Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.