Engineering Applications of Artificial Intelligence 140 (2025) 109687

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

A Multi-scale Patch Mixer Network for Time Series Anomaly Detection

Qiushi Wang [a,b], Yueming Zhu [a,b,*], Zhicheng Sun [a,b], Dong Li [a,b], Yunbin Ma [c]

[a] Key Laboratory of Networked Control Systems, Chinese Academy of Sciences, Shenyang, 110016, China
[b] Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, 110016, China
[c] PipeChina Institute of Science and Technology, Langfang, 065000, China

ARTICLE INFO

Keywords:
Lightweight
Multi-scale
Time series anomaly detection
Fully connected layer

ABSTRACT

With the development of Internet of Things (IoT) technology, a large amount of data with temporal characteristics is collected and stored. How to efficiently and accurately identify anomalies from these data is a major challenge. At present, there are many problems in the application of anomaly detection, including non-stationary data, complex and difficult-to-collect anomalies, the need for real-time detection and the limitation of computing resources. But few methods can comprehensively consider these issues. To overcome these challenges, we propose a lightweight neural network, Multi-scale Patch Mixer Network (MP-MixerNet). It is mainly composed of a Mixer Block based on fully connected layer design, which contains a Temporal-Mixer and a Spatial-Mixer, and can simultaneously model the intra- and inter-series dependencies of multivariate time series. We also perform multi-scale patch segmentation based on frequency analysis, which helps the model extract robust features from multiple period views. In addition, we design an Input Stabilization module to help the model deal with data distribution shift. Experimental results on a public time series anomaly detection dataset show that we are able to achieve higher comprehensive performance with fewer parameters and inference time.

1. Introduction

Time series anomaly detection is an important branch in the field of data analysis. It mainly focuses on identifying and locating outliers in time series data, that is, those observations that deviate significantly from normal patterns. It can help detect and respond to various key issues, so it has been widely studied in many fields such as equipment maintenance, energy consumption analysis and spacecraft telemetry (Pota et al., 2023; Song et al., 2022; Copiaco et al., 2023; Cuéllar et al., 2024).

Early scholars tried to identify anomalies by statistical, density or clustering, such as isolation forest (Liu et al., 2008), Local Outlier Factor (LOF) (Breunig et al., 2000)and k-nearest neighbor (KNN) (Teng, 2010). However, multivariate time series often involve temporal variables of multiple channels, the limitations of such methods are obvious: they can neither consider the interdependence between multiple variables nor capture the time-varying features of the series.

With the development of deep learning, neural networks have demonstrated powerful analytical and data fitting capabilities. Some scholars have begun to analyze time series by designing special neural networks. Recurrent Neural Networks (RNNs) are widely used due to their ability to effectively capture the time dependence in series data. For example, ECG-NET (Roy et al., 2023), LSTM-VAE (Park et al., 2018) and LSTM-AE (Wei et al., 2023) both utilize long short-term memory (LSTM) designed self-encoders for learning and reconstructing normal signals. With the success of the self-attention mechanism (Vaswani et al., 2017), Transformer-based network designs have shown excellent performance (Jeong et al., 2023; Kim et al., 2023; Song et al., 2024; Min et al., 2024; Wu et al., 2023). However, for well-known reasons, both RNN-based and Transformer-based models are computationally expensive, as they require a large number of learnable parameters and complex calculations and are difficult to train. In addition, they do not take into account the complex nonlinear relationships that often exist between multiple time series. Methods such as GDN (Deng and Hooi, 2021), FuSAGNet (Han and Woo, 2022), and GTA (Chen et al., 2021) use graph neural networks (GNNs) to try to model the relationships between these variables. But these methods use static time windows to construct relationships between variables, ignoring the correlation between series on different time scales and not considering the problem of variable dependency changes in short time periods.

In addition, most scholars ignore the problem of insufficient data collection which may exist in practical applications. The dataset used for training are often collected from normal systems (Goh et al., 2017; Ahmed et al., 2017). However, there is a fact that the system may have multiple operating conditions or the sensors themselves may have

[*] Corresponding author at: Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, 110016, China.
E-mail address: zhuyueming@sia.cn (Y. Zhu).

https://doi.org/10.1016/j.engappai.2024.109687
Received 6 June 2024; Received in revised form 4 November 2024; Accepted 14 November 2024
Available online 29 November 2024

Table 1
A summary statistics of dataset distributions.

Sensor	Train		Test		Rate	
	Mean	Variance	Mean	Variance	Mean	Variance
1_AIT_002_PV	0.31	0.00	0.32	0.03	+4.38%	+3462.01%
2_MCV_101_CO	0.11	0.02	0.13	0.03	+17.56%	+41.84%
2_MV_501_STATUS	0.75	0.06	0.74	0.06	−2.05%	−0.1%
2B_AIT_002_PV	0.02	0.00	146.03	1.78E4	+7.65E5%	+9.02E10%
3_LT_001_PV	0.22	0.05	0.25	0.06	+10.00%	+19.82%

numerical shift, the collected data may not cover the distribution of all normal data. The model may be biased in real operation, leading to a large number of false alarms. We statistics the data distribution of the sensors of WADI dataset, including the mean and variance, and show some representative ones, as shown in Table 1. The last two columns (rate) of the table calculate the rate of change of the mean and variance of these sensors. It can be found that although the distribution of most sensors is relatively stable, there are still some sensors whose data distribution shift greatly (such as 2B_AIT_002_PV).

To solve these problems, we propose a lightweight time series anomaly detection algorithm, Multi-scale Patch Mixer Network (MP-MixerNet). In order to deal with unpredictable data distribution shift, we designed an Input Stabilization module to improve the stability of the model. In addition, we design Mixer Block based on MLP-Mixer (Tolstikhin et al., 2021), which can use fully connected layers to achieve intra- and inter-series modeling. We also use Fast Fourier Transform (FFT) to help us convert the original series into multi-scale patch representations, and achieve modeling inter-variable dependencies in multiple time period scales to capture richer feature representations. The main contributions of our work are as follows:

(1) We propose a lightweight time series anomaly detection algorithm, MP-MixerNet, which simplifies the network structure and achieves an increase in inference speed through a network design based on MLP-Mixer, as well as an Input Stabilization module to solve the challenge of data distribution shift.
(2) We designed a Mixer Block, which realize multi-scale data fusion analysis and effectively improves the performance of the model by fusing frequency domain information.
(3) Through extensive experiments on real-world datasets, we provide empirical evidence that our approach can achieve higher performance with fewer parameters and less inference time.

The outline of this paper is as follows. Section 2 introduce the related work on time series. Section 3 describes the proposed method. Section 4 presents the results of experiments on commonly used public datasets. Section 5 summarizes the contribution of this paper and future developments.

2. Related work

In this section, we review the use of deep learning in time series anomaly detection. Also, since our approach focuses on lightweight network design, lightweight methods for time series analysis are summarized.

2.1. Deep learning for time series anomaly detection

Current deep learning methods for time series anomaly detection mainly include reconstruction-based and prediction-based methods. The reconstruction-based approach defines difficult-to-reconstruct targets as anomalies by reconstructing the inputs. LSTM-VAE (Park et al., 2018) uses LSTM and Variable Auto-Encoder (VAE) to model the multi-dimensional signal feature distribution and then reconstructs the signal using the expected feature distribution. MEMTO (Song et al., 2024) designed a gated memory module to enhance the query updating process of Transformer, and utilizes a two-stage training paradigm and a detection standard based on two-dimensional deviations to improve the performance of model reconstruction capability and anomaly detection. AnomalyBERT (Jeong et al., 2023) treats temporal data as natural language data and employs BERT models for analysis and reconstruction. PeFAD (Xu et al., 2024a) proposes a Federated Anomaly Detection framework for data feature engineering and reconstruction using linear layers and the pre-trained large language model GPT2.

Prediction-based methods use historical series to predict future series and define the portion of the prediction that has a large error as an anomaly. Kim et al. (2023) stack the outputs of multi-layer Transformers and use one-dimensional convolution for prediction, making full use of the hidden features of the intermediate layers. GDN (Deng and Hooi, 2021) uses the attention function over the neighboring sensors in the graph to learns to predict the sensor future behavior. GTA (Chen et al., 2021) builds a Multi-branch attention mixed model to learn the global bidirectional graph structure involving all variables for perform single-step time prediction. ACGSL (Pang et al., 2024) designs a series aggregation module to filter out noise disturbance, and then performs graph construction and time series prediction coding based on feature accumulation.

There are also a number of methods that combine reconstruction and prediction. CAN (Xia et al., 2023) combines adaptive graph learning methods with graph attention to learn the global and local correlation representation of variables, and develops a multilevel encoder-decoder architecture to reconstruction and prediction. HybridAD (Lin et al., 2023) uses VAE and maximum likelihood estimation to reconstruct and predict data distributions respectively.

2.2. Lightweight methods in time series

In real systems, the occurrence of abnormalities often means the need for rapid emergency response and troubleshooting. It is therefore imperative that research be conducted on methods of lightweight network design or those targeting real-time systems.

There already exist some real-time time-series anomaly detection methods. Liu et al. (2023) proposes a parallel deep network-based fault diagnosis method to maintain a very low-computational load and effectively solves the anomaly detection problem of beam pumping units. Wu et al. (2022a) proposes a Local Trend Inconsistency (LTI) and an efficient detection algorithm, and proves the possibility of parallelization for further speedup. Song et al. (2022) improves GoogleNet by improving the activation function and establishing suitable full connections, and proposes a simple anomaly detection method suitable for edge intelligence. Meta TSD-GRU (Li et al., 2022) introduces a powerful time series prediction model for predicting parking space occupancy, which greatly reduces the prediction error and improves the model training speed through rational model design. FITS (Xu et al., 2024b) uses interpolating reconstruction of the complex frequency domain for time series anomaly detection, achieving a lightweight design of the network.

With the proposal of MLP-Mixer (Tolstikhin et al., 2021), new ideas for lightweight network design are pointed out. MLP-Mixer is a new architecture proposed for computer vision tasks entirely based on FC, and has achieved performance comparable to Vision Transformers (ViT). Some recent studies have used it for time series related tasks and demonstrated its capabilities in time series modeling. TSMixer (Ekambaram et al., 2023) use MLP-Mixer for Hybrid channel modeling for multivariate prediction. Solar-mixer (Zhang et al., 2023) proposed an end-to-end solar power generation prediction model, which has achieved success in long-term prediction of photovoltaic power generation. PatchAD (Zhong et al., 2024) uses four distinct MLP Mixers and innovative dual project constraint module to build a self-supervised anomaly detection algorithm for fast and accurate anomaly detection. However, in the current time series analysis, mixer-based design has not received enough attention and research, and there is still a lot of space for research.

3. Methodology

Our method consists of three parts: Input Stabilization, Mixer Block, and Prediction Deviation Scoring. Specifically, we split the dataset into fixed-size time windows and use the Input Stabilization module to stabilize the data, then extract data features through multiple Mixer Blocks in series, and finally predict the data at the next moment by a predictor. If the error between the predicted value and the true value is too large, it indicates that there may be anomalies.

3.1. Problem statement

Assume that we have collected time series data from a normal working system containing S sensors in T_{train} timestamps $D = \{d_1, d_2, \ldots, d_{T_{train}}\}, d_{t_{train}} \in \mathbb{R}^{S\times 1}$. Our model is trained by unsupervised methods and attempts to model normal behavior in it. It is then applied to a test set with the same S sensors, which contains both normal and abnormal data. The goal of the model is to detect anomalous timestamps in the test set, where we denote 0 and 1 for normal and anomalous, respectively.

3.2. Input stabilization

In the real world, the distribution of time series data is affected by natural (week, season, etc.) or human (control strategy, operation mode, etc.) factors, resulting in a large shift in data distribution. However, it is difficult to collect data covering all distributions, so the training set may not contain all data distributions, causing the model to fail in some cases.

Recently, the RevIN (Kim et al., 2021) method has achieved success in time series forecasting tasks by stabilizing the series, but few scholars have discussed the application of series stabilization in anomaly detection. Therefore we would discuss the difference between series stabilization in prediction and our task and design a Input Stabilization module. Although our model is a prediction-based method, there are still two differences compared with prediction methods.

First, the true predicted value is visible to the model because our goal is to determine whether the predicted timestamp is an anomaly, not to actually predict it. Therefore, we adjust the scope of stabilization and concatenate the predicted timestamp with the historical time series to perform synchronous normalization instead of operating only on the historical series. Specifically, instance normalization is used on a time series of length $w+1$, where the time window size is w and the prediction length is 1. The mathematical expression is:

$$\mu = \frac{1}{w+1}\sum_{i=1}^{w+1} x_i \qquad \sigma^2 = \frac{1}{w+1}\sum_{i=1}^{w+1}(x_i-\mu)^2 \tag{1}$$

$$x' = \frac{(x-\mu)}{\sqrt{\sigma^2+\epsilon}} \tag{2}$$

where μ, $\sigma \in \mathbb{R}^{S\times 1}$. We removed the affine learning term as it has proven to be redundant in Liu et al. (2022).

Second, we do not need to get the true value of the predicted frame, we just need to compare the difference. The de-normalize term is removed and only the instance normalization term is retained for input stabilization.

3.3. Mixer block

We propose a Mixer Block based on MLP-Mixer (Tolstikhin et al., 2021) and redesign it for time series from the part of Block Structure, Temporal-Mixer and Spatial-Mixer. Specifically, the block's structure is redesigned to improve the focus on time series information, while inspired by TimesNet (Wu et al., 2022b) a multi-scale patch fusion Spatial-Mixer is designed to enhance the feature representation of the model.

Block Structure: The overall architecture of Mixer Block is shown in Fig. 1(a). It is mainly composed of two components, Temporal-Mixer and Spatial-Mixer. We reverse the cross-location operation-first design in MLP-Mixer because we are more interested in temporal feature. The effectiveness of this design is experimented and discussed in Section 4.5.

Temporal-Mixer: An LN layer and an MLP layer is used along the time dimension to extract time series information for each variable and a short connection is used to add the input to the output, as shown in Fig. 1(b):

$$TempOut = \mathrm{MLP}(\mathrm{LN}(x')) + x' \tag{3}$$

where $\mathrm{LN}(*)$ denote a Layer Normalization (LN) layer. $\mathrm{MLP}(*)$ is a multi-layer perceptron layer, shown as Fig. 1(d), which contains two Fully-Connected (FC) layers, a GLUE layer and a Drop Out layer. Note that the input does not require Per-patch Fully-connecte processing like MLP-Mixer.

Spatial-Mixer: We hope to fully exploit the correlations between multivariate time series at different time scales and obtain features that are sufficient to accurately describe them. Thus, a new Spatial-Mixer is designed to realize multi-scale feature fusion analysis. As shown in Fig. 1(c), we use a Multi-scale Patching to obtain series patches at different time scales. Then a parameter-sharing MLP is used to extract features from multiple patches along the variable dimension. Finally, the features of multiple scales are fused and added to the input using a short connection. Next we will describe this component in further detail.

For time series data, both the temporal and frequency domains contain a lot of useful information, so we want to make the best use of both in our modeling process. Thus, we use the FFT to analyze the period of the data as a basis for multi-scale patch division:

$$A = \mathrm{Avg}(\mathrm{Amp}(\mathrm{FFT}(\tilde{x}))) \tag{4}$$

where $\mathrm{FFT}(*)$ and $\mathrm{Amp}(*)$ respectively denote the calculation of FFT and amplitude. $\tilde{x}$ is the output of Temporal-Mixer. $A \in \mathbb{R}^{w}$ is the amplitude of each frequency, which is averaged from S dimensions by $\mathrm{Avg}(*)$. Then the most important K frequencies are selected according to the magnitude and calculate their periods:

$$F = \arg\mathrm{Topk}(A), \quad P = [\frac{w}{F}] \tag{5}$$

where $F = \{f_1, f_2, \ldots, f_K\}$, $P = \{p_1, p_2, \ldots, p_K\}$ denote the most important K frequencies and periods, respectively, and K is an adjustable parameter.

Based on periods P, we divide $\tilde{x}$ into multiple subset-level patches to obtain frequency-based multi-scale feature expression.

$$\tilde{x}_k = \mathrm{Reshape}(\mathrm{Padding}(\tilde{x})) \tag{6}$$

where $\tilde{x}_k \in \mathbb{R}^{S\times p_k\times f_k}$ is the result of reshape of the series based on p_k, S is the dimension of feature. $\mathrm{Padding}(*)$ uses zeros to pad the series along the time dimension so that the length of the series can fit into the reshape.

Then we use an MLP to process the variables along their dimensions:

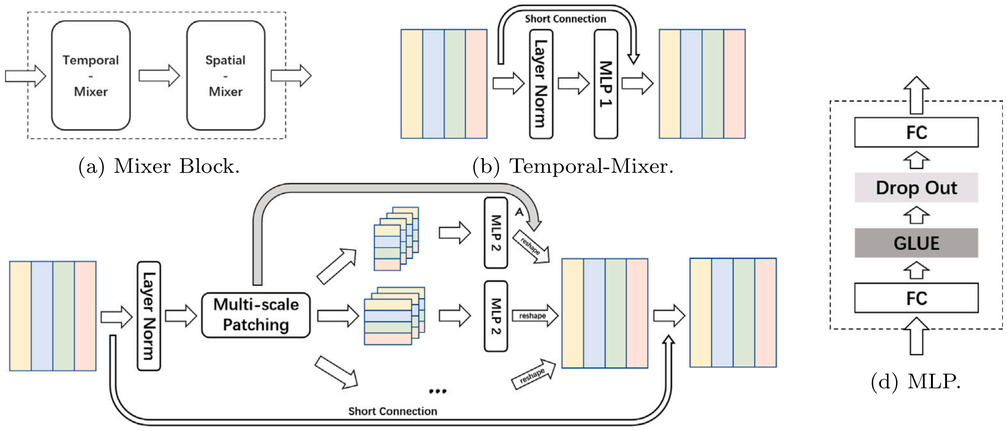
$$\tilde{x}'_k = \mathrm{MLP}(\tilde{x}_k) \qquad \tilde{x}^{spat}_k = \mathrm{Splitting}(\mathrm{Rashape}(\tilde{x}'_k)) \tag{7}$$

it is necessary to ensure that the input $\tilde{x}_i$ and the output $\tilde{x}'_i$ of the MLP have the same dimensions. We then reshape the output to size $S\times(p_i * f_i)$, and split off the padding to get a feature vector of original size.

Since the magnitude obtained from FFT calculation can reflect the importance of the corresponding frequency, we calculate the multi-scale feature aggregation weights based on the magnitude A.

$$A' = \mathrm{Softmax}(A\times\omega_A) \qquad x_{agg} = \sum_{i=1}^{K} A'_k \times \tilde{x}'_k \tag{8}$$

where, we scale the raw amplitudes using a learnable parameter w_A, and then use $\mathrm{Softmax}(*)$ to map them into the zone from 0 to 1



(a) Mixer Block.

(b) Temporal-Mixer.

(c) Spatial-Mixer.

(d) MLP.

Fig. 1. The design of Mixer Block: (a) the overall architecture of the Mixer Block; (b) the architecture of the Temporal-Mixer; (c) the architecture of the Spatial-Mixer; (d) the architecture of the MLP.

Table 2
A summary statistics of datasets.

Dataset	Features	Train	Test	Anomaly rate (%)
MSL	55	58 317	73 729	10.27
PSM	25	129 784	87 851	27.76
SWaT	51	496 800	449 919	11.98
WADI	127	1 048 571	172 801	5.99

and make them sum to 1 as the weights of the multi-scale feature aggregation. Finally, use a short connection to output the result.

$$SpatOut = x_{agg} + TempOut \tag{9}$$

3.4. Prediction deviation scoring

We use a predictor to predict the data at next timestamp t. The predictor is constructed by an MLP, whose structure is the same as Fig. 1(d). Then compare the predicted value with the true value and calculate the loss:

$$loss = \frac{1}{T_{Train} - w - 1} \sum_{t=w+1}^{T_{Train}} \| S_{pred}^{t} - S_{true}^{t} \| \tag{10}$$

where S_{pred}^{t} is the predicted value of sensors at timestamp t and S_{true}^{t} is the true value of sensors at timestamp t.

At the time of testing, we used the largest error of all the sensors as an anomaly score to highlight the impact of anomalous regions (Deng and Hooi, 2021):

$$Err_i(t) = \| S_{pred\_i}^{t} - S_{true\_i}^{t} \| \tag{11}$$

$$score = \max_{i} Err_i(t) \tag{12}$$

where $Err_i(t)$ is the error of sensor i at time t.

4. Experiments

4.1. Datasets

We evaluated our MP-MixerNet on four publicly available datasets: Mars Science Laboratory dataset (MSL) (Hundman et al., 2018), Pooled Server Metrics (PSM) (Abdulaal et al., 2021), Secure Water Treatment Testbed (SWaT) (Goh et al., 2017), and Water Distribution Testbed (WADI) (Ahmed et al., 2017), and the statistical data are shown in Table 2.

The MSL dataset is provided by NASA and contains one-hot encoding of command information collected from real spacecraft. The PSM dataset is collected from multiple application service nodes of eBay, which covers 13 weeks of training data and 8 weeks of test data. The SWaT dataset is collected from a modern industrial control system used for water treatment experiments and consists of 7 days of normal operation and 4 days of operation data with simulated attacks. The simulated attacks include cyber and physical attacks. The WADI dataset is collected from a water distribution experimental platform, which is a Cyber Physical System consisting of 127 sensors. In the next experiment, we divided the training set into training and validation sets in a ratio of 8:2.

4.2. Baselines and evaluation metrics

To prove the superiority of our model, we compare our model with the following methods:

LOF (Breunig et al., 2000) considers points with low local density as anomalies.

Isolation Forest (Liu et al., 2008) considers points that are more sparse compared to other points as anomalous.

LSTM-VAE (Park et al., 2018) uses LSTM as encoder and decoder for VAE and anomaly detection using reconstruction error. Hyperparameter setting: window size: 10; the number of hidden units in the LSTM-based encoder: 64; the number of hidden units in the LSTM-based decoder: 128.

OmniAnomaly (Su et al., 2019) combines Gated Recurrent Unit (GRU) with VAE and use reconstruction error for anomaly detection. Hyperparameter setting: window size: 100; the number of hidden units of GRU: 500.

VAEAT (He et al., 2024) uses adversarial training to enhance the LSTM-VAE effect. Hyperparameter setting: window size: 15; the dimension size of the latent variable: 400; number of encoder layers: 1; number of decoder layers: 2.

Anomaly Transformer (Xu et al., 2021) uses Transformer model with Anomaly-Attention mechanism and proposes Association-based Anomaly Criterion for anomaly detection. Hyperparameter setting: window size:100; layers of Anomaly Transformer: 3; channel number of hidden states: 512; the number of heads: 8.

TranAD (Tuli et al., 2022) is trained offline using two-stage adversarial training and meta-learning and diagnosed online using reconstruction errors. Hyperparameter setting: window size: 10; number of layers in transformer encoders: 1; number of layers in feed-forward unit of encoders: 2; hidden units in encoder layers: 64.

MEMTO (Song et al., 2024) uses memory to guide the Transformer for reconstruction and uses the error for anomaly detection. Hyperparameter setting: window size: 100; the number of memory item: 10.

GDN (Deng and Hooi, 2021) uses graph attention networks for structure learning and predicts the next moment of data, using prediction error for anomaly detection. Hyperparameter setting: length of embedding vectors: 64/64/64/128; TopK:15/10/15/30; hidden layer dimension:64/64/64/128; window size:5.

GTA (Chen et al., 2021) uses Gumbel-softmax sampling algorithm to learn the graph structure and use Influence Propagation convolution to predict the next moment of data and use the prediction error for anomaly detection. Hyperparameter setting: window size: 60; length of embedding vectors: 128; the number of heads of multi-head attention mechanism: 8; encoder layers:3; decoder layers:2; hidden layer dimension:128.

PatchAD (Zhong et al., 2024) uses Dual Project Head to learn feature mapping and considers data with large feature differences obtained from two Project Heads as anomalous. Hyperparameter setting: number of encoder layers:3; hidden layer dimension: 40; patch size: [3, 5]; window size: 105.

FITS (Xu et al., 2024b) uses frequency-domain complexes and complex neural networks to reconstruct the input after downsampling, detecting anomalies by reconstruction errors. Hyperparameter setting: window size: 200; downsample rate: 4; number of neural network layers: 1.

ACGSL (Pang et al., 2024) uses GIN for graph modeling of local signals to capture the difference between normal and abnormal signals and to detect anomalies. Hyperparameter setting: hidden layer dimension: 32; window size: 10; Top-K: 5/5/30/30.

We use precision (Pre), recall (Rec) and F1 score (F1) as evaluation metrics. In addition based on previous work (Song et al., 2024; Chen et al., 2021; Wu et al., 2022b) we use point adjustment strategy. In this approach, if we detect any anomaly during the anomaly occurrence, it will be considered as accurate detection.

4.3. Experiment set

We used Torch 1.12.1 and CUDA 11.8 for development and experiments, all experiments are performed on a single RTX 3090. The model is trained using the Adam optimizer with the learning rate initialized to 0.0001. We train the model for 30 epochs and use an early stopping strategy with a patience of 5. We set top K to 3 and layer number to 4 for all experiments and set window size to 45/90/105/105 and the hidden layer dimension of MLP to 128/64/128/256 for datasets MSL/PSM/SWaT/WADI.

4.4. Experiment results

Table 3 shows the Pre, Rec and F1 of our proposed method for anomaly detection on the four datasets in comparison with the state-of-the-art methods in recent years. We also calculate the average F1 of each method on all data in the last column (AVG) to indicate the comprehensive performance of the model, and it can be seen from the table that our model has the strongest comprehensive performance.

The best F1 in each dataset is highlighted by bolding and the next best ones are underlined, and the performance improvement of our model is recorded in parentheses after the F1 of our model. From the Table 3, it is obvious to draw the following conclusions: (1) Machine learning methods (LOF, Isolation Forest) detect anomalies based on data distribution and do not learn the deep features of time series, resulting in low performance. (2) These methods (LSTM-VAE, OmniAnomaly) are based on RNN design and can effectively analyze time series, so the performance is improved, but the relationship between variables is ignored. VAEAT combines LSTM with attention mechanism and uses two-stage training based on adversarial thinking. It can effectively capture the dependency between sequences, but ignores the changes in variable correlations at different time scales, so it cannot handle more complex data sets well. (3) Transformer-based methods (Anomaly Transformer, MEMTO, TranAD) do not consider the relationship between variables, but benefit from the powerful fitting ability of self-attention, so they achieve competitive performance. (4) Among the GNN-based methods, GDN and GTA neglect variable-dependence changes in the short time segment. ACGSL takes this change into account and uses LSTM to perform dynamic modeling based on historical states, but the time scale is still an integer multiple of the time window and cannot adaptively perform scale segmentation. (5) In lightweight models: FITS uses complex neural networks to analyze complex values in the frequency domain. Although it has only one layer of complex neural networks, it has achieved competitive results and can handle continuous value time series data well, but it has poor feature capture ability for binary information, so it performs poorly on MSL. PatchAD develops an innovative dual projection constraint module and achieves higher performance using self-supervised learning. However, the use of a fixed window size in multi-scale segmentation does not take into account frequency domain information. (6) There is a large data distribution shift in the sensor 2B_AIT_002_PV in the test set and training set of the WADI dataset, as shown in Fig. 4(c). Since most models cannot adapt to this shift, the performance is greatly reduced. At the same time, the larger number of sensors compared to other datasets also brings other challenges to anomaly detection. (7) Our method uses Input Stabilization module to ensure that the model can adapt to data distribution shift. In addition, the model can perform adaptive multi-scale patch representation based on frequency domain information and obtain a new feature representation with more accurate inter-variable dependencies through multi-scale feature fusion, thereby achieving advanced performance.

In addition, in order to verify the advancement in our network design, we counted the number of parameters and inference time of the more advanced method. In the table, we statistic the number of parameters of the model (Param) and the inference time (Time). For the sake of fairness, the statistical method of inference time is: use the model to infer 100 tensors with size $1 \times w \times N$, and take the average value of 5 experiments, where w is the time window, which we uniformly set to 100 here, and N is the number of variables. In order to visualize the comprehensive performance of the model, we show these statistics in the form of a picture, as shown in Fig. 2. The colors in the figure are used to represent different methods, the shapes are used to represent different datasets, and the sizes are used to represent inference time. The horizontal coordinate is the number of model parameters and the vertical coordinate is the F1 of the model. For easy observation, we took the average of the inference time, number of parameters, and F1 on the four datasets for display. Obviously, models that are more upper left and smaller are usually better, as they have higher performance and lighter designs.

From the Table 4 we can conclude the following: (1) The Transformer-based model has a large number of parameters and a long inference time, which is due to the complexity of self-attention calculation. Although TranAD and MEMTO have simplified from different perspectives, their inference time is still long. (2) Among the GNN-based methods, the main architecture of GDN and ACGSL has only one graph convolution layer, so the number of parameters is relatively small. In addition, both GDN and GTA contain complex self-attention calculations, so it is natural that the inference time is long. At the same time, these methods all include a graph construction process, which requires a large amount of additional information to be stored, especially ACGSL needs to store and maintain historical information. However, these parameters are not learnable parameters in neural networks and are therefore difficult to count. (3) Although VAEAT has more parameters, its inference time is short. This is because its model only contains three LSTM layers, the structure is simple, and

Table 3
Comparison with existing methods on real-world publicly available datasets.

Method	MSL			PSM		
	Pre (%)	Rec (%)	F1 (%)	Pre (%)	Rec (%)	F1 (%)
LOF	47.72	85.25	61.18	57.89	90.49	70.61
Isolation Forest	42.31	73.29	53.64	76.09	92.45	83.48
LSTM-VAE	85.49	79.94	82.62	73.62	89.92	80.96
OmniAnomaly	89.02	86.37	87.67	88.39	74.46	80.83
VAEAT	88.25	98.80	93.23	95.78	96.74	96.26
Anomaly Transformer	91.13	90.12	90.63	96.81	98.63	97.71
MEMTO	92.07	96.76	94.36	97.46	99.23	98.34
TranAD	90.38	99.99	**94.94**	92.62	99.74	96.05
GDN	85.73	87.27	86.49	95.11	77.02	85.12
GTA	91.04	91.17	91.11	93.87	62.24	74.85
ACGSL	90.05	97.33	93.55	98.58	92.26	95.32
PatchAD	89.83	86.93	88.36	97.72	98.52	98.11
FITS	61.38	80.16	69.52	97.20	90.43	93.69
MP-MixerNet (ours)	92.24	97.33	94.72(−0.22)	97.72	99.83	**98.77**(+0.43)

Method	SWaT			WADI			AVG
	Pre (%)	Rec (%)	F1 (%)	Pre (%)	Rec (%)	F1 (%)	F1 (%)
LOF	72.15	65.43	68.62	5.11	11.84	7.14	51.89
Isolation Forest	49.29	44.95	47.02	19.91	46.12	27.81	52.99
LSTM-VAE	76.00	89.50	82.20	99.47	12.82	22.71	67.12
OmniAnomaly	81.42	84.30	82.83	31.58	65.41	42.60	73.48
VAEAT	93.63	93.50	93.57	96.66	70.78	81.72	91.20
Anomaly Transformer	85.85	100.00	92.39	80.36	53.59	64.30	86.26
MEMTO	94.18	97.54	95.83	96.69	31.70	47.75	84.07
TranAD	97.60	69.97	81.51	35.29	82.96	49.51	88.89
GDN	99.35	68.12	80.82	97.50	40.19	56.92	77.34
GTA	74.91	96.41	83.69	74.56	90.50	81.76	82.85
ACGSL	90.57	99.72	94.89	76.02	85.32	80.40	91.04
PatchAD	90.03	86.88	88.43	80.31	91.65	85.60	90.13
FITS	91.74	100.0	95.69	75.16	84.62	79.61	84.63
MP-MixerNet (ours)	95.28	97.36	**96.31**(+0.48)	92.13	89.13	**90.61**(+5.01)	**95.10**(+3.90)

Table 4
Statistics of Parameter Number and Inference Time.

Method	MSL		PSM		SWaT		WADI	
	Param	Time	Param	Time	Param	Time	Param	Time
Anomaly Transformer	4 863 055	3.39	4 801 585	3.13	4 854 859	3.24	5 010 583	3.28
MEMTO	5 955 182	2.97	5 862 962	2.77	5 942 886	2.81	6 176 510	3.07
TranAD	261 243	2.93	57 273	2.98	225 519	2.98	1 352 979	3.04
GDN	14 849	1.56	12 929	1.54	14 849	1.56	47 105	1.58
GTA	844 523	3.50	775 693	3.57	838 847	3.71	1 172 051	3.86
ACGSL	47 620	0.74	47 560	0.71	47 612	0.71	47 764	0.71
VAEAT	3 850 800	0.66	3 850 800	0.43	3 850 800	0.61	3 850 800	1.13
PatchAD	284 909	1.92	217 805	1.76	270 325	1.87	610 757	2.72
FITS	2600	0.29	2600	0.29	2600	0.29	2600	0.31
MP-MixerNet (ours)	216 253	0.58	177 013	0.44	209 357	0.56	427 933	0.69

CUDA may have optimized the classic operator LSTM. (4) Among the lightweight methods, the inference time of the PatchAD method is greatly affected by the number of features and is longer on WADI. In addition, since it contains a dual-header architecture, inference can theoretically be further optimized through parallel computing. The FITS method uses frequency domain and complex neural networks for reconstruction, which greatly reduces the number of model parameters and has a faster inference speed thanks to the design of a single-layer neural network. However, due to its limited usage scenarios (only applicable to continuous data), a more complex network design is necessary. (5) Thanks to the MLP-Mixer-based architecture design, our method has very small parameters and inference time. And because our method has independent multi-scale feature calculation processes, parallel computing can theoretically be used to further shorten the inference time.

4.5. Ablation studies

To study the effectiveness of our design, we performed ablation experiments on three datasets, the results are shown in Table 5. The "Reverse Mixer" variant exchanges the positions of the two mixers in the Mixer Block. This means that the spatial-Mixer is passed first, followed by the temporal-Mixer, which is also the practice of the original MLP-Mixer (Tolstikhin et al., 2021). However, it leads to a significant decrease in model performance, probably because spatial encoding first may cover up part of the temporal information, which is more useful for anomaly detection. The "w/o ln" variant removes the LN layer from the Mixer Block, which affects the model performance to a greater extent, MLP-Mixer has the same conclusion. The "w/o multi-scale" variant removes the multi-scale branch in the Multi-scale Patching component and retains only the best scale (i.e. Top k=1). The results show that the multi-scale design can help the model extract features more effectively from multiple period scales, which can greatly improve the model performance. The "w/o Stabilization" variant, which removes the Input Stabilization module, also leads to a significant decrease in model performance. Especially on the WADI dataset, the Input Stabilization module can help the model solve the problem of data distribution shift and greatly improve the model performance.

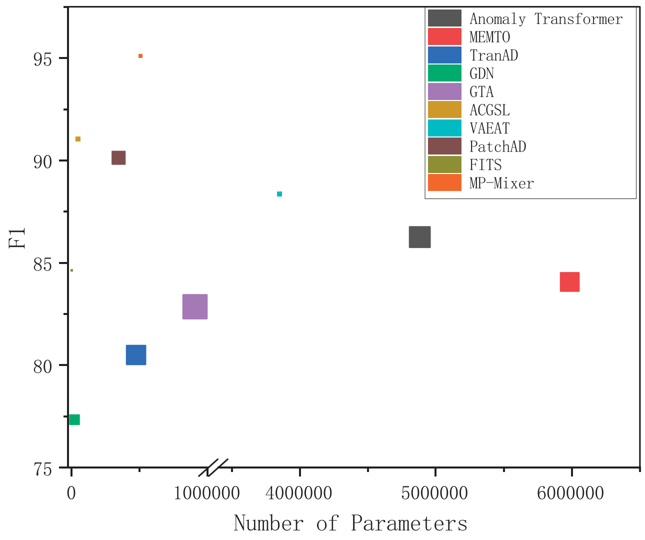


Fig. 2. Comparison of the number of parameters.

Table 5
Ablation studies result.

Method	F1		
	MSL	SWaT	WADI
Ours	94.72	96.31	90.61
Reverse Mixer	91.28	90.69	71.35
w/o ln	92.60	82.56	69.09
w/o multi-scale	92.33	94.87	84.55
w/o Stabilization	83.59	94.39	59.98

4.6. Visual analysis

To more visually illustrate how the MP-MixerNet works, we generated a time series of data containing anomalies, as shown in Fig. 3. This is a behavior-driven of time series anomaly, including point- and pattern-wise anomaly accordingly (Lai et al., 2021).

Fig. 3(a) shows synthetic data containing anomalies, with the anomaly portion boxed and the anomaly type labeled next to it. Fig. 3(b) and Fig. 3(c) respectively represent the results of the original signal (blue line) and model prediction (green line) without Input Stabilization module and processed with Input Stabilization module. Their difference is mainly reflected in the Trend anomaly. A new steady state is formed after the Trend anomaly, but the distribution of the new steady state may not be seen during model training. Input stabilization module can help the model adapt to the new steady state. Fig. 3(d) shows the anomaly score using input stabilization module. It can be found that all types of anomalies are detected. However, it is also observed that when just leaving the pattern-wise abnormal state, the model's predictions and anomaly scores fluctuate. This is because the historical time window required for prediction contains too many outliers. Furthermore, point-wise anomaly has little impact on model predictions because the number of outliers is much smaller than the time window.

In order to further demonstrate the performance of our algorithm on real datasets, we performed a visual analysis on the WADI dataset, as shown in Fig. 4, where the red part is the abnormal time period and the green part is the normal time period. From Fig. 4(a) and Fig. 4(b), it can be seen that our algorithm is able to identify system anomalies sensitively at the early stage of anomalies. Fig. 4(c) is the sensor in the dataset where a large distributional shift has occurred between the training and test sets, which may be due to the system running on conditions where the model has not been trained. Although the anomaly score was affected at the beginning of the shift, the model quickly adapted to the new data distribution and worked properly.

4.7. Sensitivity analysis

Fig. 5 shows the impact of MP-MixerNet's hyperparameters on different datasets. We can find that: first, the addition of Mixer Block can help the model capture deeper features and improve model performance, but the improvement is not obvious after a certain depth. Second, increasing k can improve model performance, but too large k has a negative effect because it may introduce too much useless period information. Third, increasing window size can improve model performance because it is closely related to the period. These conclusions can provide strong support for the practical application and deployment of the model in the future.

5. Conclusion

We propose an innovative time series anomaly detection algorithm, MP-MixerNet, which is designed based on the MLP-Mixer architecture to achieve intra- and inter-series feature extraction as well as model simplification. At the same time, the design of multi-scale feature fusion based on frequency domain information further enhances the feature expression capability. In addition, the proposed Input Stabilization module can help MP-MixerNet adapt to the shift of data distribution and make the model more robust. Through a large number of experiments to analyze the comprehensive performance, parameter amount, and inference speed of the model, MP-MixerNet is able to achieve higher performance with very few parameters and inference time. In the future, we plan to evaluate MP-MixerNet in more real-world scenarios and improve its interpretability. In addition, we hope to further improve the inference speed of the model with the help of parallel computing. Finally, we would like to apply it on other time series tasks (e.g., prediction and classification, etc.) to improve its generalization performance.

CRediT authorship contribution statement

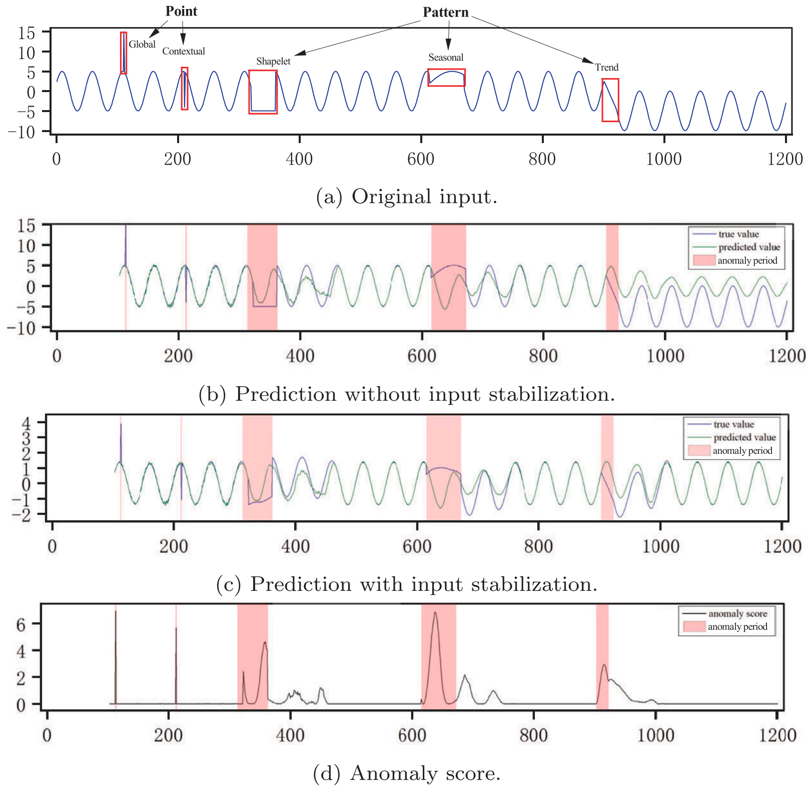
Qiushi Wang: Writing – review & editing, Validation, Methodology, Investigation, Funding acquisition. **Yueming Zhu:** Writing – original draft, Visualization, Software, Methodology, Conceptualization. **Zhicheng Sun:** Visualization, Validation, Methodology, Data curation. **Dong Li:** Validation, Formal analysis, Data curation. **Yunbin Ma:** Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding

This work was supported in part by the Applied Basic Research Project of Liaoning Province of China [grant numbers 2023JH2/101300183]; and the Science and Technology Plan Major Project of Liaoning Province of China [grant numbers 2022JH1/10400009].



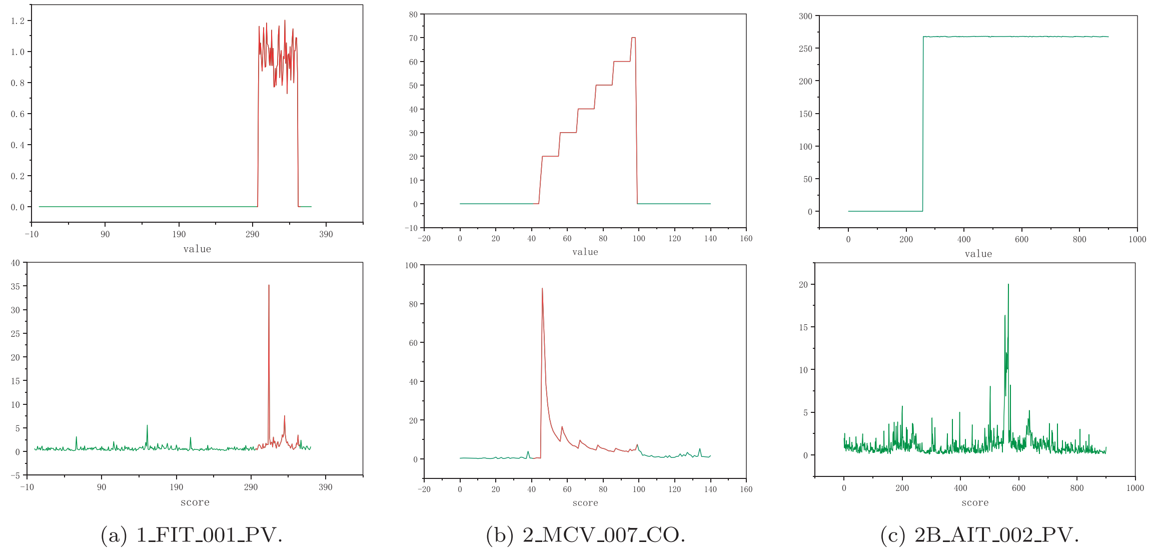
(a) Original input.

(b) Prediction without input stabilization.

(c) Prediction with input stabilization.

(d) Anomaly score.

Fig. 3. Visual analysis.

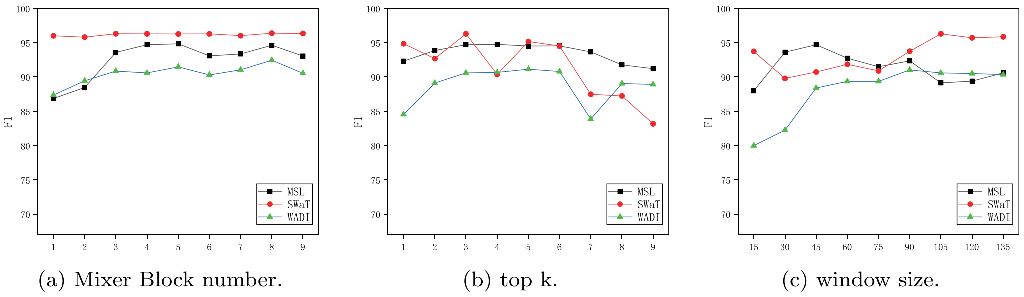


(a) 1_FIT_001_PV.

(b) 2_MCV_007_CO.

(c) 2B_AIT_002_PV.

Fig. 4. Visual analysis on WADI: values of _FIT_001_PV, 2_MCV_007_CO and 2B_AIT_002_PV (top) and system anomaly scores for that time period (bottom).



(a) Mixer Block number.

(b) top k.

(c) window size.

Fig. 5. Sensitivity analysis.

Data availability

Authors in this article have used publicly available dataset.

References

Abdulaal, A., Liu, Z., Lancewicki, T., 2021. Practical approach to asynchronous multivariate time series anomaly detection and localization. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. pp. 2485–2494.

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. pp. 25–28.

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. pp. 93–104.

Chen, Z., Chen, D., Zhang, X., Yuan, Z., Cheng, X., 2021. Learning graph structures with transformer for multivariate time-series anomaly detection in IoT. IEEE Internet Things J. 9 (12), 9179–9189.

Copiaco, A., Himeur, Y., Amira, A., Mansoor, W., Fadli, F., Atalla, S., Sohail, S.S., 2023. An innovative deep anomaly detection of building energy consumption using energy time-series images. Eng. Appl. Artif. Intell. 119, 105775.

Cuéllar, S., Santos, M., Alonso, F., Fabregas, E., Farias, G., 2024. Explainable anomaly detection in spacecraft telemetry. Eng. Appl. Artif. Intell. (ISSN: 0952-1976) 133, 108083.

Deng, A., Hooi, B., 2021. Graph neural network-based anomaly detection in multivariate time series. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, (5), pp. 4027–4035.

Ekambaram, V., Jati, A., Nguyen, N., Sinthong, P., Kalagnanam, J., 2023. Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 459–469.

Goh, J., Adepu, S., Junejo, K.N., Mathur, A., 2017. 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 11. Springer, pp. 88–99.

Han, S., Woo, S.S., 2022. Learning sparse latent graph representations for anomaly detection in multivariate time series. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 2977–2986.

He, S., Du, M., Jiang, X., Zhang, W., Wang, C., 2024. VAEAT: Variational AutoeEncoder with adversarial training for multivariate time series anomaly detection. Inform. Sci. 120852.

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. pp. 387–395.

Jeong, Y., Yang, E., Ryu, J.H., Park, I., Kang, M., 2023. Anomalybert: Self-supervised transformer for time series anomaly detection using data degradation scheme. arXiv preprint arXiv:2305.04468.

Kim, J., Kang, H., Kang, P., 2023. Time-series anomaly detection with stacked transformer representations and 1D convolutional network. Eng. Appl. Artif. Intell. 120, 105964.

Kim, T., Kim, J., Tae, Y., Park, C., Choi, J.-H., Choo, J., 2021. Reversible instance normalization for accurate time-series forecasting against distribution shift. In: International Conference on Learning Representations.

Lai, K.-H., Zha, D., Xu, J., Zhao, Y., Wang, G., Hu, X., 2021. Revisiting time series outlier detection: Definitions and benchmarks. In: Thirty-Fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1).

Li, J., Qu, H., You, L., 2022. An integrated approach for the near real-time parking occupancy prediction. IEEE Trans. Intell. Transp. Syst. 24 (4), 3769–3778.

Lin, W., Wang, S., Wu, W., Li, D., Zomaya, A.Y., 2023. HybridAD: A hybrid model-driven anomaly detection approach for multivariate time series. IEEE Trans. Emerg. Top. Comput. Intell..

Liu, S., Song, C., Wu, T., Zeng, P., 2023. A lightweight fault diagnosis method of beam pumping units based on dynamic warping matching and parallel deep network. IEEE Trans. Syst. Man Cybern.: Syst..

Liu, F.T., Ting, K.M., Zhou, Z.-H., 2008. Isolation forest. In: 2008 Eighth Ieee International Conference on Data Mining. IEEE, pp. 413–422.

Liu, Y., Wu, H., Wang, J., Long, M., 2022. Non-stationary transformers: Exploring the stationarity in time series forecasting. Adv. Neural Inf. Process. Syst. 35, 9881–9893.

Min, H., Lei, X., Wu, X., Fang, Y., Chen, S., Wang, W., Zhao, X., 2024. Toward interpretable anomaly detection for autonomous vehicles with denoising variational transformer. Eng. Appl. Artif. Intell. (ISSN: 0952-1976) 129, 107601.

Pang, H., Wei, S., Li, Y., Liu, T., Zhang, H., Qin, Y., Zhao, Y., 2024. Asymptotic consistent graph structure learning for multivariate time series anomaly detection. IEEE Trans. Instrum. Meas..

Park, D., Hoshi, Y., Kemp, C.C., 2018. A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. IEEE Robot. Autom. Lett. 3 (3), 1544–1551.

Pota, M., De Pietro, G., Esposito, M., 2023. Real-time anomaly detection on time series of industrial furnaces: A comparison of autoencoder architectures. Eng. Appl. Artif. Intell. 124, 106597.

Roy, M., Majumder, S., Halder, A., Biswas, U., 2023. ECG-NET: A deep LSTM autoencoder for detecting anomalous ECG. Eng. Appl. Artif. Intell. 124, 106484.

Song, J., Kim, K., Oh, J., Cho, S., 2024. Memto: Memory-guided transformer for multivariate time series anomaly detection. Adv. Neural Inf. Process. Syst. 36.

Song, C., Liu, S., Han, G., Zeng, P., Yu, H., Zheng, Q., 2022. Edge-intelligence-based condition monitoring of beam pumping units under heavy noise in industrial internet of things for industry 4.0. IEEE Internet Things J. 10 (4), 3037–3046.

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. pp. 2828–2837.

Teng, M., 2010. Anomaly detection on time series. In: 2010 IEEE International Conference on Progress in Informatics and Computing, vol. 1, IEEE, pp. 603–608.

Tolstikhin, I.O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., Uszkoreit, J., et al., 2021. Mlp-mixer: An all-mlp architecture for vision. In: Advances in Neural Information Processing Systems, vol. 34, pp. 24261–24272.

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.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30.

Wei, Y., Jang-Jaccard, J., Xu, W., Sabrina, F., Camtepe, S., Boulic, M., 2023. LSTM-autoencoder-based anomaly detection for indoor air quality time-series data. IEEE Sens. J. 23 (4), 3787–3800.

Wu, Y., Dong, Y., Zhu, W., Zhang, J., Liu, S., Lu, D., Zeng, N., Li, Y., 2023. CLformer: Constraint-based locality enhanced transformer for anomaly detection of ancient building structures. Eng. Appl. Artif. Intell. (ISSN: 0952-1976) 126, 107072.

Wu, W., He, L., Lin, W., Su, Y., Cui, Y., Maple, C., Jarvis, S., 2022a. Developing an unsupervised real-time anomaly detection scheme for time series with multi-seasonality. IEEE Trans. Knowl. Data Eng. 34 (09), 4147–4160.

Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., Long, M., 2022b. Timesnet: Temporal 2d-variation modeling for general time series analysis. In: The Eleventh International Conference on Learning Representations.

Xia, F., Chen, X., Yu, S., Hou, M., Liu, M., You, L., 2023. Coupled attention networks for multivariate time series anomaly detection. IEEE Trans. Emerg. Top. Comput. 12 (1), 240–253.

Xu, R., Miao, H., Wang, S., Yu, P.S., Wang, J., 2024a. PeFAD: A parameter-efficient federated framework for time series anomaly detection. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 3621–3632.

Xu, J., Wu, H., Wang, J., Long, M., 2021. Anomaly transformer: Time series anomaly detection with association discrepancy. In: International Conference on Learning Representations.

Xu, Z., Zeng, A., Xu, Q., 2024b. FITS: Modeling time series with $10k$ parameters. In: The Twelfth International Conference on Learning Representations.

Zhang, Z., Wang, J., Xia, Y., Wei, D., Niu, Y., 2023. Solar-mixer: An efficient end-to-end model for long-sequence photovoltaic power generation time series forecasting. IEEE Trans. Sustain. Energy.

Zhong, Z., Yu, Z., Yang, Y., Wang, W., Yang, K., 2024. PatchAD: Patch-based MLP-mixer for time series anomaly detection. arXiv preprint arXiv:2401.09793.