Multi-Scale Anomaly Detection for Time Series with Attention-based Recurrent Autoencoders

Qingning Lu QNLU@SMAIL.NJU.EDU.CN
Wenzhong Li LWZ@NJU.EDU.CN
Chuanze Zhu ZHUCZ@SMAIL.NJU.EDU.CN
Yizhou Chen CHENYIZHOU@SMAIL.NJU.EDU.CN
Yinke Wang YINKEWANG@SMAIL.NJU.EDU.CN
Zhijie Zhang 13505242562@163.COM
Linshan Shen SHENLS@SMAIL.NJU.EDU.CN
Sanglu Lu SANGLU@NJU.EDU.CN
Department of Computer Science and Technology, Nanjing University, Nanjing

Editors: Emtiyaz Khan and Mehmet Gönen

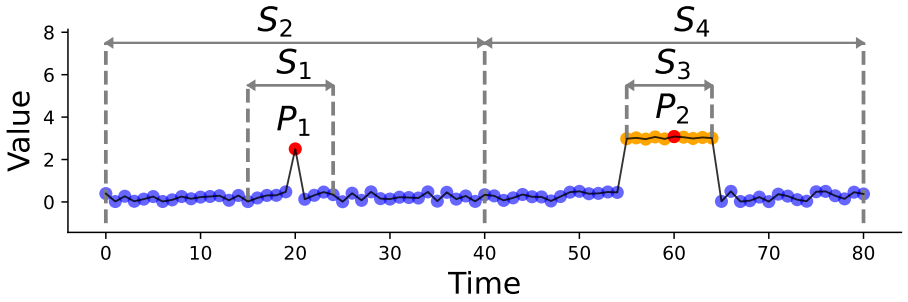
Abstract

Anomaly detection on time series is an important research topic in data mining, which has a wide range of applications in financial markets, biological data, information technology, manufacturing system, etc. However, the existing time series anomaly detection methods mainly capture temporal features from a single-scale viewpoint, which cannot detect multi-scale anomalies effectively. In this paper, we propose a novel approach of Multi-scale Anomaly Detection for Time Series (MAD-TS) with an attention-based recurrent autoencoder model to solve the above problem. The proposed method adopts a hierarchically connected recurrent encoder to extract the features of a time series from different levels. The multi-scale features are then fused by a hierarchical decoder with attention mechanism to reconstruct the original sequence at different scales. Based on the reconstruction errors at multiple scales, anomaly scores can be learned for different data points, which can be used to infer the anomaly status of the time series. Extensive experiments based on five open time series datasets show that the proposed MAD-TS method achieves significant performance improvement on anomaly detection compared to the state-of-the-arts.

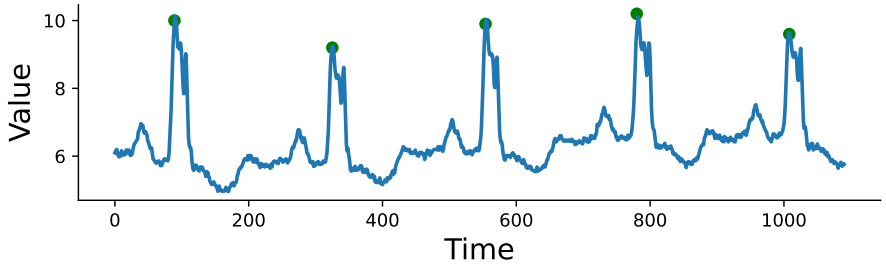
Keywords: Anomaly detection; Multi-Scale; Time Series.

1. Introduction

Nowadays modern information systems typically generate a large amount of data in the form of time series, which include the continuous measurement of key performance indicators (KPIs), the reports from stock markets, the sensors' records from the cyber-physical systems (CPSs), etc. Time series anomaly detection tends to identify abnormal status in each time step from the time series data, which has become an important topic in data mining Chalapathy and Chawla (2019). Time series anomaly detection has a wide range of applications in the fields of risk assessment in financial markets, automatic operation in IT systems, failure detection in complex industrial systems, etc. Due to the fact that anomalies are often rare in real-world time series and labeling anomalies from time series



(a) Isolated noise and contextual anomalies in time series.



(b) Seasonal patterns in time series.

Figure 1: Time series anomalies from different scale's viewpoints.

is time-consuming and expensive in practice, time series anomaly detection is typically formulated in the unsupervised learning setting Su et al. (2019); Zhang et al. (2019).

In the past decades, a large amount of work has been performed in the area of unsupervised time series anomaly detection, which can be categorized as discrimination-based, forecasting-based, and reconstruction-based methods. Discrimination-based methods measure the similarity between two sequences and distinguish abnormal patterns from normal ones using statistical methods or early machine learning models such as Local Outlier Factor Breunig et al. (2000), one-class SVM Ma and Perkins (2003), EM Pan et al. (2010), etc. Forecasting-based methods build a predictive model based on historical values and decide whether an observed value is abnormal by calculating its prediction error, which typically includes ARMA Wold (1938), LSTM Hochreiter and Schmidhuber (1997), HTM Ahmad et al. (2017), etc. Reconstruction-based methods learn a compact representation of normal data by minimizing the difference between reconstruction and the input series, which are popularly applied with the modern deep learning models such as AutoEncoder (AE) Zong et al. (2018), LSTM-VAE Park et al. (2018), Generative Adversarial Network (GAN) Audibert et al. (2020), etc.

Despite the great efforts of unsupervised methods for time series, it is still challenging for them to model the complex nonlinear temporal dynamics from a comprehensive viewpoint. While most existing unsupervised methods either capture temporal features of time series from a fixed time window or tend to reconstruct a whole time series session unfocusedly and indiscriminately, they mainly extract single-scale features and may not be able to detect anomalies effectively from the following scenarios. (1) *Isolated noise*. As the time series

data illustrated in Fig. 1(a)subfigure, if observed from a small scale S1, point P1 could be considered as an anomaly by most unsupervised methods since its value is significantly different from that of the other points. However, if observed from a larger scale S2, P1 probably is an isolated noise since the time series has normal values before and after that point. (2) *Contextual anomalies*. As shown in Fig. 1(a)subfigure, if observed from the scale S3, point P2 could be considered normal by some fixed time-window methods since it has similar value as its nearby points. However, if observed from scale S4, P2 and all points in S3 should be contextual anomalies. (3) *Seasonal patterns*. Time series data typically have seasonal patterns occurring at regular intervals. As illustrated in Fig. 1(b)subfigure, the time series has periodical patterns and some peaks indicated by green dots occur in every period regularly which should not be considered abnormal. However, the recurrent neural networks (RNN) based methods Kieu et al. (2019); Shen et al. (2021) fail to capture such seasonal patterns if their lookback length is too short. (4) *Error accumulation*. While deep generative models are popular for time series in recent years, they used reconstruction error to compute anomaly scores and perform poorly on long time series due to error accumulation in reconstructing a long sequence. Since the calculation of each step in the decoding process depends on the result of the previous step, it will cause accumulative errors in the final results.

In this paper, we propose a novel Multi-scale Anomaly Detection method for Time Series called MAD-TS to address the above issues. It introduces a hierarchically connected recurrent encoder to project the input time series into latent feature representations from different temporal scales. Then the multi-scale features are fused by a hierarchical decoder with attention mechanism to reconstruct the original sequence at different scales. Based on the reconstruction errors, anomaly scores can be learned for different data points to infer the anomaly status of the time series. The performance of the proposed method is verified by extensive experiments.

The contributions of this paper are summarized as follows.

- We propose the novel idea of detecting time series anomalies from a multi-scale viewpoint, which was rarely studied in the literature.
- We introduce a recurrent autoencoder-decoder structure with attention mechanism to capture comprehensive temporal features from different scales, which can effectively detect anomalies and alleviate error accumulation for long time series.
- We conduct extensive experiments based on five open time series datasets, which show that the proposed MAD-TS method achieves significant performance improvement on time series anomaly detection compared to the state-of-the-arts.

2. Related Work

Time series anomaly detection has been extensively studied in the past decades. We summarize the existing works into three categories: discrimination-based methods, forecasting-based methods, and reconstruction-based methods. In addition, we surveyed the existing works that consider multi-scale features for anomaly detection and discussed their shortcomings.

2.1. Discrimination-based Methods

Earlier time series anomaly detection works were usually based on statistical methods, which assumed most of the time series are normal while a few are anomalous. Measuring the similarity between two sequences and further distinguishing abnormal patterns were the key problems. Solutions include clustering: k-Means Nairac et al. (1999), EM Pan et al. (2010), one-class SVM Ma and Perkins (2003), etc. Local Outlier Factor Breunig et al. (2000) was an algorithm for finding anomalous data points by measuring the local deviation of a given data point with respect to its neighbours. Ruff et al. (2018) introduced Deep Support Vector Data Description (Deep SVDD) which was trained on an anomaly detection based objective. Shen et al. (2020) proposed Temporal Hierarchical One-Class (THOC) network which utilized multiple hyperspheres obtained with a hierarchical clustering process for anomaly detection.

2.2. Forecasting-based Methods

The forecasting-based methods build a predictive model based on historical values and determine whether an observed value is abnormal by calculating its prediction error. Autoregressive moving average (ARMA) Wold (1938) built a parametric model of the time series, which was widely used in a number of fields. Autoregressive integrated moving averaged (ARIMA) Moayedi and Masnadi-Shirazi (2008) allowed for the management of nonstationarity by adding a number of differencing steps during the processing phase to move the data toward a more stationary distribution. Ahmad et al. (2017) proposed a streaming data anomaly detection algorithm based on Hierarchical Temporal Memory (HTM). In recent years, deep learning methods such as Long Short-Term Memory (LSTM) Hochreiter and Schmidhuber (1997) have shown good performance on time series prediction. Malhotra et al. (2015) used stacked LSTM networks for anomaly detection in time series by training on non-anomalous data and using the multi-step prediction errors to evaluate the likelihood of anomalous behavior. Hundman et al. (2018) proposed a complementary unsupervised and nonparametric anomaly thresholding approach for detecting spacecraft anomalies.

2.3. Reconstruction-based Methods

The reconstruction-based methods learn a compact representation of normal data by minimizing the difference between reconstruction and the input series. AutoEncoder (AE) is the most commonly used reconstruction model. Zong et al. (2018) presented a Deep Autoencoding Gaussian Mixture Model (DAGMM) for unsupervised anomaly detection which utilized a deep autoencoder to generate a low-dimensional representation and reconstruction error for each input data point and further fed into a Gaussian mixture model. Xu et al. (2018) proposed Donut, an unsupervised anomaly detection algorithm based on variational autoencoder in order to perform seasonal KPIs with various patterns and data quality. Audibert et al. (2020) proposed a fast and stable method called UnSupervised Anomaly Detection for multivariate time series (USAD) based on adversely trained autoencoders. Park et al. (2018) introduced a long short-term memory based variational autoencoder (LSTM-VAE) that fused signals and reconstructs their expected distribution. Malhotra et al. (2016) proposed a LSTM-based Encoder-Decoder scheme for Anomaly Detection (EncDec-AD) that

learned to reconstruct normal time-series behavior, and thereafter used reconstruction error to detect anomalies. Su et al. (2019) proposed OmniAnomaly, a stochastic recurrent neural network for multivariate time series anomaly detection that worked well robustly for various devices. Zhang et al. (2019) proposed Multi-Scale Convolutional Recurrent Encoder-Decoder (MSCRED), which constructed multi-resolution signature matrices and fed them into a convolutional LSTM network. Kieu et al. (2019) exploited autoencoders built using sparsely-connected recurrent neural networks aiming to reduce the effects of some autoencoders being overfitted to outliers. Shen et al. (2021) propose a recurrent network ensemble called Recurrent Autoencoder with Multiresolution Ensemble Decoding (RAMED) that used decoders with different decoding lengths and a coarse-to-fine fusion mechanism. Li et al. (2021) proposed InterFusion, an unsupervised method that simultaneously modeled the inter-metric and temporal dependency for multivariate time series.

2.4. Multi-Scale Features for Anomaly Detection

Previous works on time series anomaly detection had rarely considered multi-scale features of time series. Zhang et al. (2019) simply concatenated signature matrices of different lookback lengths together as input of the convolutional encoder without feature fusion in the encoding-decoding process. Shen et al. (2021) used decoders of different lengths with the encoding process using an ensemble method. The ensemble method filtered out some noises but did not effectively extract multi-scale features. Shen et al. (2020) used multi-resolution encoders and fused them with a multi-level one-class neural network. Wang et al. (2021) used two independent RNN modules to model global and local features and used a weighted loss for balancing, which can capture different levels of sequential patterns simultaneously in discrete event sequences. There is still lack of works on multi-scale time series feature extraction and fusion for unsupervised anomaly detection, which is the major focus of this paper.

3. Problem Formulation

Given a time series $\mathcal{X} = [\mathbf{x}_1, ..., \mathbf{x}_N]$ of length N, where $\mathbf{x}_t \in \mathbb{R}^M$ $(1 \leq t \leq N)$ is a M-dimensional vector, and each dimension represents the observed value of a metric at time step t. $\mathcal{X}$ is called univariate time series when $M = 1$, otherwise multivariate time series. A time series sequence can be divided into a number of sub sequences using a sliding window with lookback length T, which is represented by $[\mathbf{X}_T, ..., \mathbf{X}_{N-1}, \mathbf{X}_N]$, where $\mathbf{X}_t = [\mathbf{x}_{t-T+1}, ..., \mathbf{x}_{t-1}, \mathbf{x}_t]$ is a subsequence with length T.

For unsupervised time series anomaly detection, $\mathcal{X}$ is used as the input for model training, which is assumed to contain only normal patterns. Given an unseen observation value $\mathbf{x}_t$, the goal of anomaly detection is to assign a label $y \in \{0, 1\}$ according to how far it deviates from historical normal patterns (usually referred to as the anomaly score), where $y = 1$ means that the observation is anomalous, otherwise it is normal.

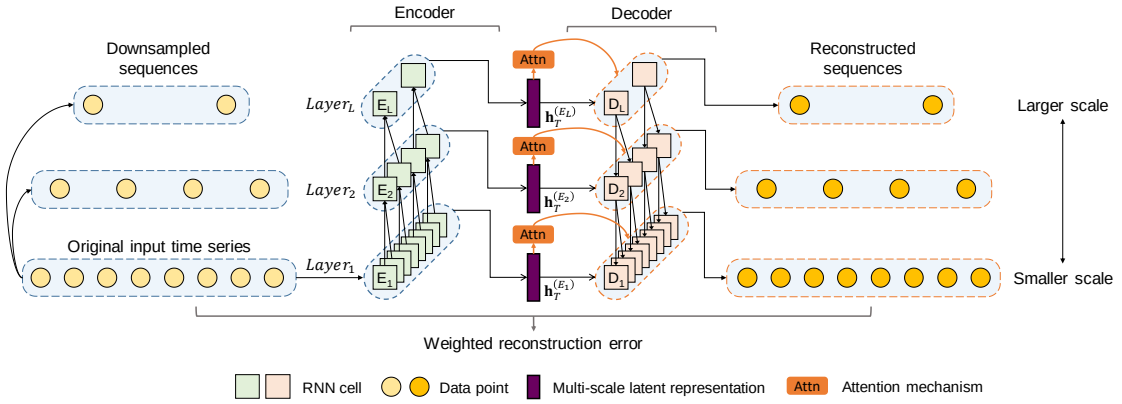


Figure 2: The proposed architecture of Multi-scale Anomaly Detection for Time Series (MAD-TS).

4. Proposed Method

4.1. Architecture

In this section, we propose a Multi-scale Anomaly Detection method for Time Series (MAD-TS), which employs a multi-scale recurrent encoder-decoder structure with attention mechanism to capture non-linear features of time series at multiple scales for anomaly detection. The proposed model is shown in Figure 2. On the left, the input time series is connected to a hierarchical recurrent neural network (RNN) encoder to extract multi-scale features, which are fused from lower-layer to higher-layer to form latent representations. On the right, the multi-scale features are fed to a hierarchical RNN-decoder, which fuses the features from higher-layer to lower-layer with attention mechanism to obtain the reconstruction results at different scales of the original sequence. The model is trained with normal historical time series and then applied for anomaly detection online. When an unseen sequence is fed to the model, the mean reconstruction error is computed, which is used as the anomaly score for anomaly detection. We introduce the detailed mechanisms of the encoder and decoder in the following.

4.2. Multi-Scale RNN Encoder for Feature Fusion

In order to better capture multi-scale features in time series, we adopt a hierarchically connected recurrent neural network (RNN) structure as the encoder. As illustrated in Fig. 2, the encoder consists of L layers, with the lowest layer E_1 using the original time series as input, and the other layers $[E_2, ..., E_L]$ use the output of s corresponding units' hidden states from the lower layer as input, where s is called the *fusion stride* of the hierarchical model. At a given time step t, for a particular encoding layer E_i with index i, the hidden state of the RNN unit is denoted by:

$$\mathbf{h}_t^{(E_i)} = \begin{cases} f^{(E_i)}([\mathbf{x}_t; \mathbf{h}_{t-1}^{(E_i)}]), & \text{if } i = 1 \\ f^{(E_i)}([\hat{\mathbf{h}}_t^{(E_i)}; \mathbf{h}_{t-1}^{(E_i)}]), & \text{o/w} \end{cases} \tag{1}$$

where $f^{(E_i)}$ is a RNN cell, such as LSTM or GRU, and $\hat{\mathbf{h}}_t^{(E_i)}$ is the multi-scale feature fusion input, which is defined as the average of the hidden states of s units in the lower layer:

$$\hat{\mathbf{h}}_t^{(E_i)} = \frac{1}{s} \sum_{j=0}^{s-1} \mathbf{h}_{s \times t-j}^{(E_{i-1})}. \tag{2}$$

The intuition of using a hierarchical multi-scale RNN encoder for feature fusion is that the higher-level RNN units can capture macro-scale temporal features, while the lower-level RNN units can capture micro-scale temporal features. For the origin time series of length T, the encoding length of a certain layer is s times that of the previous layer. After multi-scale fusion encoding, L hidden representations of different scales of the original time series are learned by the encoder.

4.3. Attention-based Hierarchical RNN Decoder

Compared with the bottom-up encoding process, the multi-scale feature fusion in the decoding process is a top-down approach. As illustrated in Fig. 2, the ith layer of the decoder $f^{(D_i)}$ reconstructs a sequence of length $T_i = T/s^{i-1}$. For fusing coarse-grained features, we concatenate the hidden state of the previous time step and the corresponding hidden state of the previous layer together to calculate the reconstruction at the current time step. Then the reconstruction $\mathbf{y}_t^i$ and the hidden state $\mathbf{h}_t^{(D_i)}$ are used to obtain the next hidden state $\mathbf{h}_{t-1}^{(D_i)}$, which is computed by

$$\mathbf{h}_{t-1}^{(D_i)} = f^{(D_i)}([\mathbf{y}_t^i; \mathbf{h}_t^{(D_i)}]). \tag{3}$$

To deal with the problem of error accumulation during the decoding process, we introduce an attention-based mechanism Vaswani et al. (2017) for feature fusion. As shown in Fig. 2, the latent representation of each layer of the decoder is attached with an attention mechanism, which learns the attention weights to fuse the latent features from different scales with different fusion stride. The attention weights can urge the model to focus on important features and effectively alleviate error accumulation from unimportant subsequences of long time series. The reconstructed time series $[\mathbf{y}_1^i, \mathbf{y}_2^i, ..., \mathbf{y}_{T_i}^i]$ can be computed by

$$\mathbf{y}_t^i = \begin{cases} \mathbf{W}_i(concat[\mathbf{h}_t^{(D_i)}; Attn_t^i]) + \mathbf{b}_i, & \text{if } i = \text{L} \\ \mathbf{W}_i(concat[\mathbf{h}_t^{(D_i)}; \mathbf{h}_{\lceil t/s \rceil}^{(D_{i+1})}; Attn_t^i]) + \mathbf{b}_i. & \text{o/w} \end{cases} \tag{4}$$

Note that there are many ways to implement the attention mechanism, and we adopt the scaled dot-product, which can directly establish the relationship between the hidden states of the current decoding step and that of each step in the encoding process. In addition, we adopt matrix multiplication without introducing additional neural networks that can effectively reduce the amount of calculation and facilitate training. The proposed attention mechanism can be computed by

$$Attn_t^i = \sum_{j=1}^{T_i} \frac{\mathbf{h}_t^{(D_i)^T} \mathbf{h}_j^{(E_i)}}{\sqrt{\mathbf{h}_j^{(E_i)}}} \mathbf{h}_j^{(E_i)}. \tag{5}$$

Each layer of the decoder can generate a reconstructed time series of different scales. In order to make these reconstructed time series as close to the original time series as possible, we downsample the original time series to obtain L time series of different scales, and calculate the reconstruction errors between the downsampled sequences and the reconstruction results.

The goal of training the model is to minimize the mean reconstruction error (MRE) represented by the following loss function:

$$\mathcal{L}_{MRE} = \frac{1}{L} \sum_{i=1}^{L} \mathcal{L}_{MSE}^i, \tag{6}$$

$$where \quad \mathcal{L}_{MSE}^i = \sum_{t=1}^{T_i} ||\mathbf{y}_t^i - \mathbf{x}_t^i||^2. \tag{7}$$

4.4. Anomaly Score and Detection

After training the model, when an unseen time series $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_T]$ is input to the model, it detects anomaly as follows. Firstly it applies the model to reconstruct the time series at different scales to form the reconstruction series $[\mathbf{Y}_1, ..., \mathbf{Y}_L]$. Then it downsamples the original time series with stride s to form downsampling subsequences $[\mathbf{X}_1, ..., \mathbf{X}_L]$ at the same scale of reconstruction accordingly.

With the reconstruction series and downsampling series, the mean reconstruction error $\mathbf{e}_T$ can be obtained by calculating $\mathcal{L}_{MRE}$ in Eq. (6). For the whole time series, we can obtain a series of mean reconstruction errors $\mathbf{E} = [\mathbf{e}_T, ...\mathbf{e}_N]$.

We then fit the calculated $\mathbf{E}$ in the validation set to a standard normal distribution $\mathcal{N}(\mu, \mathbf{\Sigma})$ to obtain normalized anomaly scores which are calculated by

$$a(\mathbf{x}_t) = (\mathbf{e}_t - \mu)^T \mathbf{\Sigma}^{-1} (\mathbf{e}_t - \mu). \tag{8}$$

In the inference stage, a data point $\mathbf{x}_t$ is considered as anomalous if its corresponding anomaly score $a(\mathbf{x}_t)$ is greater than a predefined threshold.

5. Experiment

5.1. Experimental Setup

Datasets

The experiments are performed based on five real-world univariate and multivariate time series datasets, which are described in the following:

- ECG (electrocardiogram): This is a set of six datasets that contains anomalous beats from electrocardiograms.
- 2D-gesture (video surveilance): This dataset records the X-Y coordinates of an actor's right hand in the video.
- Power demand: This dataset contains one year's power demand at a Dutch research facility.
- WADI (water distribution): This dataset is collected from 123 sensors and actuators from an extension of the SWaT testbed. It consists of 14 days under normal operation and 2 days with attack scenarios.
- KDD99: This is a classical network anomaly detection dataset that records different types of attacks and normal connections.

The statistics of these datasets[1] are summarized in Table 1.

Dataset	# dim	# length	Anomaly (%)
ECG	2	33998	3.04
2D-gesture	2	11251	24.63
power demand	1	32931	10.39
WADI	127	957374	5.77
KDD99	34	1056408	30

Table 1: Statistics of the datasets.

Baseline Algorithms

We compare the proposed method with six competing anomaly detection baselines:

- **Local Outlier Factor (LOF)** Breunig et al. (2000), a density-based outlier detection method.
- **Isolation Forest (IF)** Liu et al. (2008), an unsupervised anomaly detection algorithm conducted by random partitioning.

1. The ECG, 2D-gesture, and power demand datasets are downloaded from https://www.cs.ucr.edu/~eamonn/discords/, WADI is from https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info/#wadi, and KDD99 is from http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

- **EncDec-AD** Malhotra et al. (2016), it utilizes an RNN structure to reconstruct the original sequence.
- **OmniAnomaly** Su et al. (2019), a state-of-the-art stochastic recurrent neural network for multivariate time series anomaly detection.
- **MSCRED** Zhang et al. (2019), it constructs multi-resolution signature matrices and further feeds them into a convolutional LSTM network to capture temporal features.
- **USAD** Audibert et al. (2020), it learns feature representations based on adversely trained autoencoders. Source codes of these baselines are downloaded from the Internet, except that USAD method was reproduced by us according to Audibert et al. (2020).

Performance Metrics

By setting a threshold on the anomaly score, we can get Precision (P), Recall (R), and F1 score of each compared algorithm:

$$P = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN}, F1 = \frac{2 \times P \times R}{P + R}, \tag{9}$$

where TP is the True Positives; FP is the False Positives; FN is the False Negatives. We search for the threshold in the feasible interval and report the result corresponding to the highest F1 score.

Implementation Details

We implemented the proposed MAD-TS[2] model with the deep learning python library PyTorch. All the experiments are performed on a machine with Intel(R) Core(TM) i7-9700 CPU @ 3.00GHz 8 cores, NVIDIA GeForce RTX 2070 SUPER, 16 GB RAM.

For all deep learning based methods, we use Adam as the optimizer with a learning rate 0.001. Time series window length T is 64 for ECG and 2D-gesture, 512 for power demand, and 80 for the others with stride 1. For the proposed model, we use LSTM as RNN unit with hidden size 10. The number of hierarchical layers L in the encoder-decoder model is set to 3, and the multi-scale fusion stride s is set to 3 by default, except that $s = 4$ for power demand. We give equal weight to the reconstruction results of each layer, which means $\alpha_i = 1$. In order to allow hyperparameter tuning, we use 10% of the training set as the validation set. We apply min-max scaling within each metric on all the datasets.

5.2. Performance Comparison

Table 2 shows the results on the ECG dataset, and Table 3 shows the results on the remaining four datasets. As can be seen, LOF and IF generally perform worse than methods based on deep learning, due to the reason that they fail to capture the non-linear temporal dependencies in the sequence. OmniAnomaly tends to achieve extreme results on ECG, while MSCRED performs roughly well on most of the datasets. USAD generally performs better than the above-mentioned methods, which obtain a sub-optimal result.

2. Source code is available at https://github.com/AlumLuther/MAD-TS.

Methods	ECG1			ECG2			ECG3		
	P	R	F1	P	R	F1	P	R	F1
LOF	45.18	38.29	41.45	94.74	45.28	61.28	62.07	12.08	20.22
IF	32.67	48.70	39.10	58.51	34.59	43.48	13.26	30.87	18.55
EncDec-AD	98.02	36.80	53.51	97.14	64.15	77.27	83.21	76.51	79.72
OmniAnomaly	31.65	32.71	32.17	49.23	60.37	54.23	18.11	32.21	23.18
MSCRED	76.19	59.26	**66.67**	100.0	62.50	76.92	84.62	73.33	78.57
USAD	88.24	39.03	54.12	100.0	67.03	80.45	84.34	46.98	60.34
MAD-TS	87.23	45.72	60.00	100.0	68.55	**81.34**	91.80	75.17	**82.66**
Methods	ECG4			ECG5			ECG6		
	P	R	F1	P	R	F1	P	R	F1
LOF	11.54	96.90	20.63	64.71	47.48	54.77	8.24	91.53	15.12
IF	21.23	29.46	24.68	24.41	37.41	29.55	8.37	99.47	15.44
EncDec-AD	19.89	85.27	32.26	42.91	89.21	57.94	15.82	28.04	20.23
OmniAnomaly	12.19	100.0	21.73	21.15	39.56	27.56	8.62	100.0	15.87
MSCRED	23.26	76.92	35.71	37.93	78.57	51.16	11.11	78.95	19.48
USAD	20.71	95.35	34.02	36.39	91.37	52.05	15.07	29.10	19.86
MAD-TS	24.51	68.22	**36.07**	52.36	71.94	**60.61**	14.52	37.57	**20.94**

Table 2: Performance on the ECG dataset. The best results are marked with bold.

Methods	2D-gesture			power demand			WADI			KDD99		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
LOF	27.82	87.21	42.18	15.29	28.13	19.81	10.46	49.94	17.29	95.38	99.55	97.42
IF	28.84	68.04	40.22	7.85	89.77	14.44	29.92	15.83	20.71	96.52	99.29	97.88
EncDec-AD	48.81	58.46	53.20	13.98	54.20	22.22	24.11	26.48	25.24	97.84	97.60	97.72
OmniAnomaly	27.70	79.67	41.11	37.11	48.60	42.08	99.47	12.98	22.96	98.49	98.85	98.67
MSCRED	61.26	59.11	60.17	55.80	34.32	42.50	19.73	29.59	23.67	97.31	99.43	98.36
USAD	56.18	55.35	55.76	53.43	42.09	47.09	26.58	26.08	26.33	98.34	99.42	98.88
MAD-TS	67.11	61.30	**64.07**	55.82	49.64	**52.55**	54.35	18.15	**27.22**	99.92	98.24	**99.07**

Table 3: Performance on four datasets: 2D-gesture, power demand, WADI and KDD99. The best results are marked with bold.

In general, MAD-TS outperforms the other methods on all datasets (except ECG1 in Table 2). This indicates that multi-scale feature with attention mechanism can effectively improve the anomaly detection results. Overall, improvements on univariate dataset (power demand) and bivariate datasets (ECG and 2D-gesture) are more significant than that on multivariate datasets (WADI and KDD99), with a 6.8% higher average F1 score on ECG dataset than that of USAD, the state-of-the-art method.

Visualization of Anomaly Scores

For a better understanding of the diverse performance of different algorithms, Figure 3 shows the original time series and the visualized anomaly scores of different algorithms on the 2D-gesture dataset, where red points indicate ground truth anomalies and blue

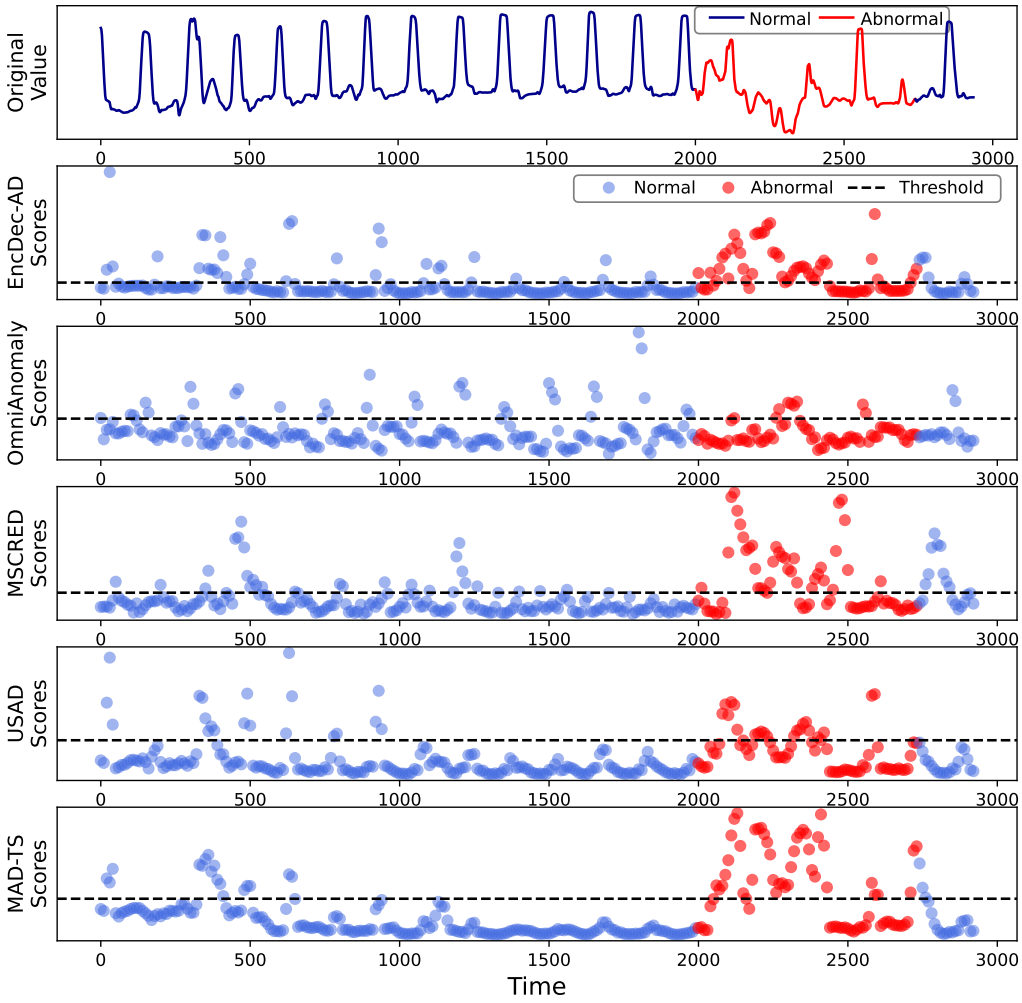


Figure 3: Visualization of anomaly scores of different algorithms on 2D-gesture dataset.

points indicate normal points. An ideal anomaly detection algorithm should make all red points above the threshold while all blue points below the threshold. Although most of the methods detect many anomalies, there are still a number of false positives which influence their overall performance in terms of F1 score. MAD-TS obtains both fewer false negatives and false positives, thus leading to better performance.

5.3. Ablation Study

Next, we conduct experiments to show the effectiveness of the proposed multi-scale feature fusion and attention mechanism. We remove the corresponding component individually and retrain the model to test the performance. The results on 2D-gesture dataset are illustrated in Table 4. It is shown that both the multi-scale feature fusion and the attention mechanism significantly improved the performance of the model by more than 5%. The combination of both can capture temporal features at different scales and effectively reduce the accumulative error during the decoding process.

w/ multi-scale feature fusion	w/ attention mechanism	P	R	F1
×	×	48.81	58.46	53.20
×	✓	61.80	56.70	59.14
✓	×	59.43	61.84	60.61
✓	✓	67.11	61.30	**64.07**

Table 4: Comparisons of variants using different network structures.

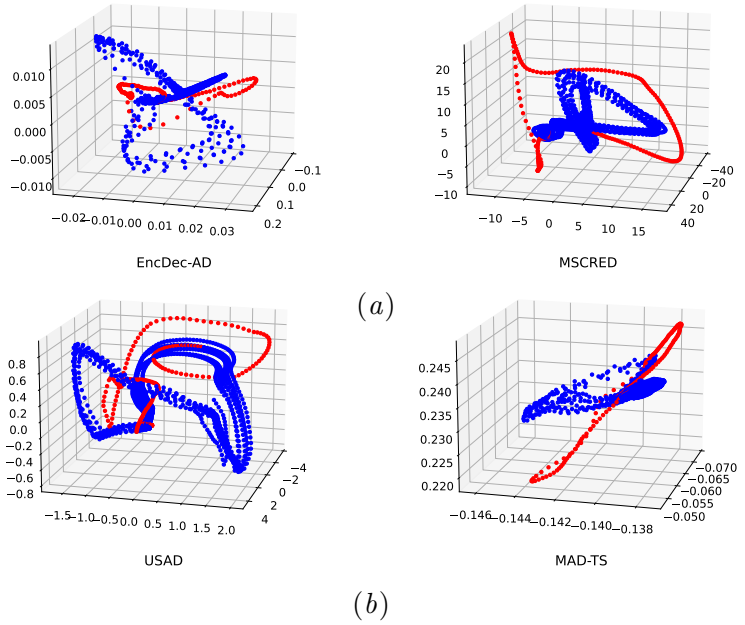


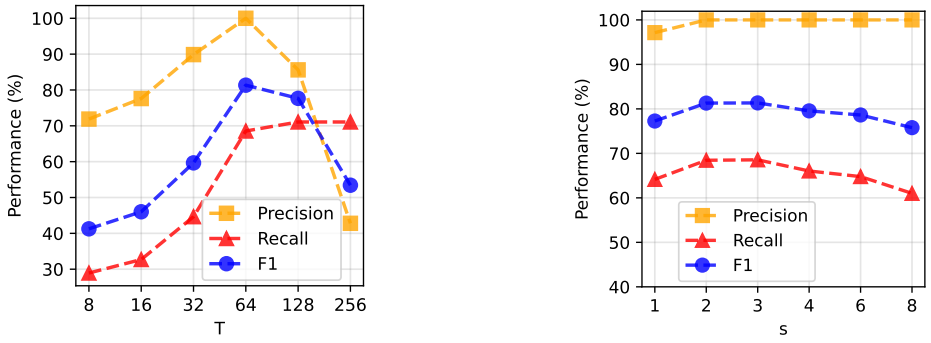
Figure 4: Visualization of latent representations learned by different algorithms.

Visualization of Latent Representations

To demonstrate the effectiveness of MAD-TS, we visualize the latent representations of different deep learning algorithms. Figure 4 shows the 3-dimensional hidden space learned by different algorithms on the ECG dataset. The red points represent features corresponding to ground truth anomalies, while the blue points represent those corresponding to normal points. As can be seen, the latent representations of anomalies learned by MAD-TS are significantly different from those of the normal points. By using a hierarchical multi-scale RNN encoder, MAD-TS can capture features of the origin time series at multiple scales. After training, the latent representations of normal data are as similar as possible and can be interpreted as a compact cluster in the hidden space.

5.4. Hyperparameter Analysis

We analyze the influence of two hyperparameters: the time series window length T and the multi-scale feature fusion stride s in the proposed framework. Experiments are carried out on one of the ECG datasets, with T varying from {8, 16, 32, 64, 128, 256} and s varying from {1, 2, 3, 4, 6, 8}. The results are shown in Figure 5.



(*a*) Varying of T (*b*) Varying of s

Figure 5: Sensitivities of hyperparameters on ECG.

Figure 5(*a*)subfigure shows the influence of window length T. As can be seen, a larger window can improve the performance effectively, especially in terms of recall. However, as T continues to increase, the precision has experienced a significant drop. Furthermore, the complexity of training and inference also increases linearly. For the purpose of balancing precision and recall, a window length sufficient to capture larger-scale features while not too large is necessary. Figure 5(*b*)subfigure shows the influence of fusion stride s. There is no feature fusion of different scales when $s = 1$, which naturally leads to a bad result. The performance begins to decline when s exceeds 4. This is because an excessively large s will cause the topmost decoding step to be too small and fail to provide effective large-scale features. This suggests that we need to choose a suitable s for a particular dataset.

6. Conclusion

This paper addresses the challenges of detecting multi-scale anomalies for time series and proposes a novel approach called MAD-TS to solve the problem. The proposed method adopted a hierarchically connected recurrent encoder to extract comprehensive features of a time series from different scales, and applied a hierarchical decoder with attention mechanism to fuse the multi-layer features and reconstruct the original sequence at different scales. The weighted reconstruction errors could be used as anomaly scores to infer the anomaly status of the time series. Extensive experiments based on five open time series datasets showed that the MAD-TS significantly outperforms the state-of-the-arts.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (Grant Nos. 61972196, 61832008, 61832005), the Collaborative Innovation Center of Novel Software Technology and Industrialization, and the Sino-German Institutes of Social Computing. The corresponding author is Wenzhong Li (lwz@nju.edu.cn).

References

Subutai Ahmad, Alexander Lavin, Scott Purdy, and Zuha Agha. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, 262:134–147, 2017.

Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A Zuluaga. Usad: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD*, pages 3395–3404, 2020.

Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000.

Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey, 2019.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *the 24th ACM SIGKDD*, pages 387–395, 2018.

Tung Kieu, Bin Yang, Chenjuan Guo, and Christian S Jensen. Outlier detection for time series with recurrent autoencoder ensembles. In *Proceedings of IJCAI-19*, pages 2725–2732, 2019.

Zhihan Li, Youjian Zhao, Jiaqi Han, Ya Su, Rui Jiao, Xidao Wen, and Dan Pei. Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding. In *Proceedings of the 27th ACM SIGKDD*, pages 3220–3230, 2021.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008.

Junshui Ma and Simon Perkins. Time-series novelty detection using one-class support vector machines. In *Proceedings of the International Joint Conference on Neural Networks, 2003.*, volume 3, pages 1741–1745. IEEE, 2003.

Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. Long short term memory networks for anomaly detection in time series. In *Proceedings*, volume 89, pages 89–94, 2015.

Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Lstm-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*, 2016.

H Zare Moayedi and MA Masnadi-Shirazi. Arima model for network traffic prediction and anomaly detection. In *2008 International Symposium on Information Technology*, volume 4, pages 1–6. IEEE, 2008.

Alexandre Nairac, Neil Townsend, Roy Carr, Steve King, Peter Cowley, and Lionel Tarassenko. A system for the analysis of jet engine vibration data. *Integrated Computer-Aided Engineering*, 6(1):53–66, Jan 1999.

Xinghao Pan, Jiaqi Tan, Soila Kavulya, Rajeev Gandhi, and Priya Narasimhan. Ganesha: Blackbox diagnosis of mapreduce systems. *ACM SIGMETRICS Performance Evaluation Review*, 37(3):8–13, 2010.

Daehyung Park, Yuuna Hoshi, and Charles C Kemp. A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. *IEEE Robotics and Automation Letters*, 3(3):1544–1551, 2018.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International conference on machine learning*, pages 4393–4402. PMLR, 2018.

Lifeng Shen, Zhuocong Li, and James Kwok. Timeseries anomaly detection using temporal hierarchical one-class network. *Advances in Neural Information Processing Systems*, 33: 13016–13026, 2020.

Lifeng Shen, Zhongzhong Yu, Qianli Ma, and James T Kwok. Time series anomaly detection with multiresolution ensemble decoding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9567–9575, 2021.

Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD*, pages 2828–2837, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Zhiwei Wang, Zhengzhang Chen, Jingchao Ni, Hui Liu, Haifeng Chen, and Jiliang Tang. Multi-scale one-class recurrent neural networks for discrete event sequence anomaly detection. In *Proceedings of the 27th ACM SIGKDD*, KDD '21, page 3726–3734, 2021.

Herman Wold. *A study in the analysis of stationary time series*. PhD thesis, Almqvist & Wiksell, 1938.

Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, et al. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In *Proceedings of the 2018 World Wide Web Conference*, pages 187–196, 2018.

Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, Cristian Lumezanu, Wei Cheng, Jingchao Ni, Bo Zong, Haifeng Chen, and Nitesh V Chawla. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1409–1416, 2019.

Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations*, 2018.