Neurocomputing 491 (2022) 261–272

Contents lists available at ScienceDirect

Neurocomputing

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

Efficient time series anomaly detection by multiresolution self-supervised discriminative network

Desen Huang [a,1], Lifeng Shen [b,1], Zhongzhong Yu [a], Zhenjing Zheng [a], Min Huang [c,*], Qianli Ma [a,d,*]

[a] School of Computer Science and Engineering, South China University of Technology, Guangzhou, China
[b] Artificial Intelligence Thrust Area, Information Hub, Hong Kong University of Science and Technology, Hong Kong Special Administrative Region
[c] School of Software Engineering, South China University of Technology, Guangzhou, China
[d] Key Laboratory of Big Data and Intelligent Robot (South China University of Technology), Ministry of Education, China

ARTICLE INFO

Article history:
Received 23 June 2021
Revised 26 December 2021
Accepted 20 March 2022
Available online 23 March 2022
Communicated by Zidong Wang

Keywords:
Self-supervised learning
Discriminative network
Time series
Anomaly detection

ABSTRACT

Time series anomaly detection aims to identify abnormal subsequences in time series that are markedly different from the temporal behaviors of the entire sequence. Although previous density-based or proximity-based anomaly detection methods are usually used for anomaly detection, they are still suffering from high computational costs due to the need of traversing the whole training dataset during testing. Recently, reconstruction-based deep learning methods are popular for time series anomaly detection. However, they may not work well because their objective is to recover all information appeared in time series, including high-frequency noises. In this paper, we propose a simple yet efficient method called **M**ultiresolution **S**elf-**S**upervised **D**iscriminative **Net**work (MS^{2}D-Net) for efficient time series anomaly detection. Specifically, the MS^{2}D-Net includes a multiresolution downsampling module, a feature extraction module, and a self-supervised discrimination module. The multiresolution downsampling module generates some multiresolution samples by downsampling the original time series with different sampling rates and creates different pseudo-labels representing multi-scale behaviors in time series. Then, in the feature extraction module, a shallow convolution network is used to extract temporal dynamics in time series at multiple resolutions. Finally, the self-supervised discrimination module uses the pseudo-labels obtained from the multiresolution downsampling module as the self-supervised information to help separate anomalies from the normal time series samples. Experimental results show that the proposed MS^{2}D-Net can outperform recent strong deep learning baselines on 18 benchmarks for time series anomaly detection with a much lower computational cost.

1. Introduction

Time series anomaly detection is an important research topic, which is closely related to our daily life [1–4]. Industry devices, such as spacecrafts [5], server machines [6,7], robot-assisted systems [8,9], engines [10], are typically monitored with multiple sensors and generate massive time series in a short time [11]. Anomalies of time series usually are the subsequences of a time series that are markedly different from the temporal behaviors in the whole time series (one example is shown in Fig. 1). In the past decades, time series anomaly detection has received widespread attention in many real-world applications. Real-world scenarios require anomaly detection algorithms to have high performance and fast execution speed such that potential risks can be avoided in time [12–15]. Take the ECG anomaly detection task [16] as an example, the discordant observation coincides with the early ventricular contraction of the ventricle. Efficient time series anomaly detection methods can remind the doctor to treat the patient in time.

Since time series anomaly detection is an important module in a wide range of industrial applications, current researchers have spared a lot of effort on developing various efficient anomaly detection techniques for time series. Existing studies on time series anomaly detection can be roughly divided into two categories: supervised methods [17–20] and unsupervised methods [21,22]. Supervised methods use "abnormal" and "normal" label information to guide training, while in unsupervised methods, only unlabeled samples are accessible. Compared to the former category,

* Corresponding author at: School of Computer Science and Engineering, South China University of Technology, Guangzhou, China (Q. Ma), and School of Software Engineering, South China University of Technology, Guangzhou, China (M. Huang).
E-mail addresses: minh@scut.edu.cn (M. Huang), qianlima@scut.edu.cn (Q. Ma).
[1] Equal contribution.

https://doi.org/10.1016/j.neucom.2022.03.048
0925-2312/

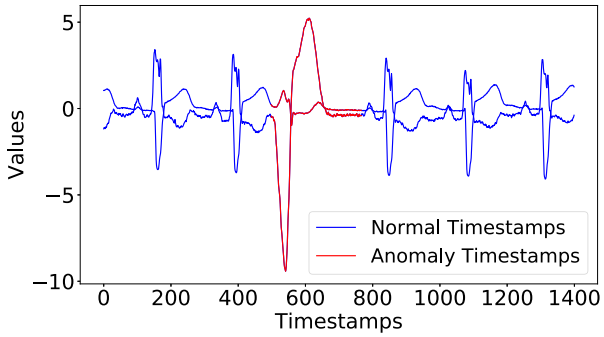


Fig. 1. A sample of ECG time series with a sub-sequence of anomalies (two dimensions).

unsupervised setting is more challenging since there are not available label information to help train the model.

Most traditional machine learning methods for anomaly detection depend on a profile of normal instances. Abnormal instances can be identified if they can not meet the normal profile [23]. These methods mainly include proximity-based methods [24] and density-based methods [25]. Both of them can be regarded as distance-based methods. Although distance-based methods are verified as a state-of-the-art anomaly detection framework [26] and show good interpretability, they still need to re-traverse the entire training set to find the nearest samples during inference that making them inefficient computationally. Especially, when dealing with long time series data, these methods always suffer high computational costs. Other methods based on classification, including OCSVM [27], learn a linear boundary to describe normality presented in data. However, these methods are not robust in time series anomaly detection tasks since the optimal boundary may be much high-dimensional and complex such that the hyper-plane or the ball-based decision boundary suffer from their limited capacities on data fitting [28].

Recently, deep learning-based methods have made significant progress in time series anomaly detection due to their advantages on representation learning. Most of them are mainly based on reconstruction (as shown in Fig. 2(a)). Specifically, they map the original input into a low-dimensional space and then reconstruct the original input from the learned low-dimensional representation. Their underlying assumption is that a good model for anomaly detection can easily describe the normal features in a low-dimensional space. Under this assumption, samples that cannot be reconstructed well will be identified as anomalies. Hence, the reconstruction error can be naturally defined as an indicator for anomaly detection.

Specifically, there are two reconstruction paradigms widely used in deep learning: i) the auto-encoding method and ii) the prediction-based method. Classical models in the former one are based on auto-encoder. Representative works include LSTM-VAE [8], SISVAE [29] and BeatGAN [22]. The prediction-based methods usually depend on the estimation of future trends, such as LSTM Predictor [30,5] and Sequence to Sequence model [10,31,32]. However, above reconstruction-based models may not work well in time series anomaly detection tasks because they aim to recover all information appeared in time series input, even some useless or noise information. Besides, the prediction-based paradigm is also limited when time series is complex and high-nonlinear due to the difficulty of modeling complex time series dynamics.

As a powerful unsupervised learning framework, self-supervised methods have been successfully applied in computer vision tasks, including object detection and segmentation [33]. Its main idea is to design efficient auxiliary tasks to help learn robust representations for given data. Popular auxiliary self-supervised tasks (e.g., classification) are based on image augmentation operations, including image rotation, pixel permutation and translation [34]. Recently, self-supervised technique has been applied in image anomaly detection tasks [35]. By introducing suitable auxiliary classification tasks (it uses different image augmentation views as its self-supervision information [36]), normal patterns in image data can be easily captured by a self-supervised discriminator. To the best of our knowledge, self-supervised methods have not yet been explored in time series anomaly detection. One of the main difficulties to directly use self-supervised techniques in time series is that we can not clearly define informative time series augmentation strategies like image transformation. Inappropriate augmentation transformation will easily damage regular temporal patterns in time series and bring additional noises to misleading model training.

In this paper, we argue that multiresolution temporal information in time series can be used a kind of self-supervision information for time series anomaly detection. Specifically, we propose a simple yet efficient model called **M**ultiresolution **S**elf-**S**upervised **D**iscriminative **Net**work (MS^{2}D-Net). The proposed model consists of three basic components: i) a multiresolution downsampling module; ii) a shallow convolution network-based feature extraction module and iii) a self-supervised discrimination module. The multiresolution downsampling module generates augmented samples at different time resolutions (by different downsampling rates). For each augmented sample, a pseudo-label is assigned by the corresponding downsampling rate. Subsequently, based on the shallow convolution-based feature extraction module, we can

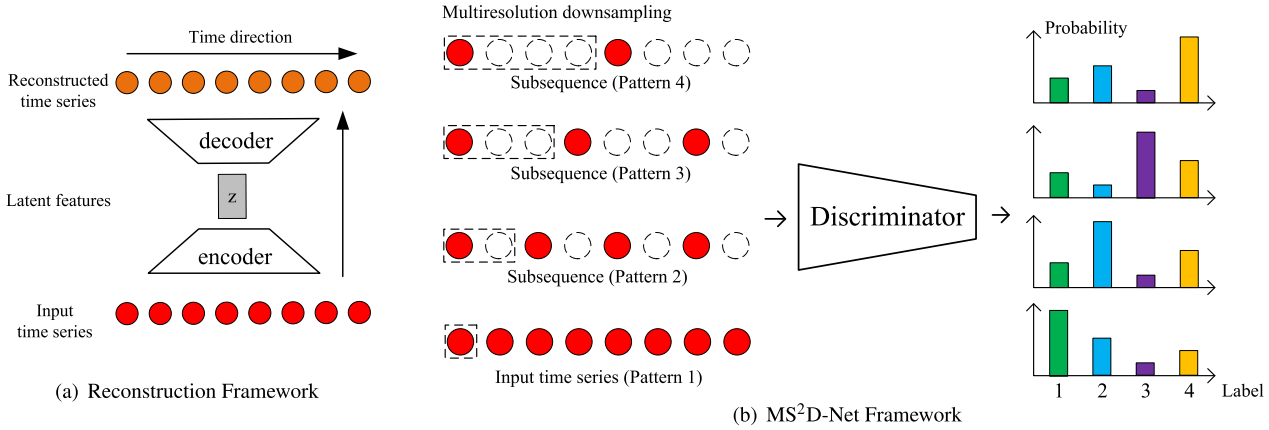


Fig. 2. Comparisons between the reconstruction framework and the proposed MS^{2}D-Net framework for time series anomaly detection..

extract temporal representations for each downsampled time series. Finally, the linear mapping-based self-supervised discrimination module can be learned in a supervised way. Intuitively, this self-supervised discrimination module aims to describe normal patterns at various time resolutions. When testing an abnormal instance, the self-supervised discrimination module will fail to discriminate the abnormal instance's patterns as its confidence of discrimination will decrease at some resolution levels. Therefore, we can use the confidence of discrimination as an indicator of anomaly score.

We highlight the main difference between the existing reconstruction framework and our proposed discriminative framework in Fig. 2. As can be seen, the reconstruction framework only captures the time series feature in the original time-ordered space. However, its performance is often limited by the fitting capacities of the selected encoder and decoder. Differently, our proposed discriminative framework describes the normality with the auxiliary multiresolution self-supervised tasks and it is without the need of reconstructing complex nonlinear time series.

In summary, our contributions mainly include:

- We propose a new model called **M**ultiresolution **S**elf-**S**upervised **D**iscriminative **Net**work (MS^{2}D-Net). Since the proposed MS^{2}D-Net does not need to autoencode or self-predict the complex nonlinear time series, it is effective for time series anomaly detection;
- A multiresolution self-supervised scheme is developed to model normal characteristics of time series at multiple time resolutions;
- The proposed model achieves the best average performance on 18 time series anomaly detection tasks with a lower computational cost. Detailed analysis on visualization, hyperparameter sensitivity and computational efficiency also verify the effectiveness of our proposed method.

The rest of this paper is organized as follows. Section 2 discusses the related work on existing time series anomaly detection methods and the most related discriminative self-supervised methods. Section 3 presents the proposed framework for time series anomaly detection formally. Section 4 describes the detailed experimental setting, reports results and analysis on 18 time series anomaly detection tasks. And finally, we make conclusions in Section 5.

2. Related Work

In this section, we briefly review related anomaly detection methods in three clues: i) traditional machine learning methods; ii) reconstruction-based deep learning methods; and iii) mostly-related discriminative self-supervised anomaly detection methods.

2.1. Traditional Machine Learning Methods

Traditional anomaly detection methods include one-class support vector machine (OCSVM) which finds a hyperplane to describe anomalies [27]. In addition, density-based and prototype-based methods such as LOF [25] and MP [24] are also commonly used for anomaly detection. They calculate the distance between a test sample and its nearest sample and then the distance can be used as an indicator of anomaly detection [24]. Moreover, ensemble methods based on decision trees are also used for anomaly detection. One of the classical models is isolation forest [23], which builds an ensemble of decision trees. Based on the average path lengths of tree, the isolation forest can identify whether a test sample is abnormal or not.

2.2. Reconstruction Methods in Deep Learning

In deep learning, reconstruction methods are often used for weakly supervised anomaly detection [37,38] and unsupervised anomaly detection [39]. In time series anomaly detection field, representative models include sequence-to-sequence network [10,31,32] and auto-encoder based networks [22,29]. They encode time series as low-dimensional features via an encoder and then map these features back to original input space to calculate a reconstruction error. The basic assumption of reconstruction methods is that the normal patterns can be captured effectively in a low-dimensional space such that the normal samples will be more likely to have a smaller reconstruction error.

Furthermore, there are some prediction-based methods (e.g., the predictive recurrent neural network [30,5]) for anomaly detection since the prediction error can be employed as the anomaly score. However, compared to auto-encoder based networks, these predictive methods are often limited to the long-horizon prediction capacity of the recurrent neural network.

2.3. Discriminative Self-supervised Methods

Self-supervised learning recently has shown its powerful ability on representation learning [33], and far surpassed the reconstruction methods in image anomaly detection tasks [35,36]. Specifically, the self-supervised anomaly detection method generates a series of augmented samples based on semantic-invariant image transformation such as rotation, pixel permutation and translation [34]. These different augmented views are used as pseudo-labels to represent the underlying self-supervised information. Then a simple discriminator can be utilized to distinguish different augmented views such that invariant underlying patterns in images can be captured well. Although the self-supervised technology has been successfully applied in image anomaly detection, it is still an open question on *how to use self-supervision learning for time series*. For time-series data, there are no explicit data-augmentation views in analogy with the image transformations (e.g., rotations). In this paper, we address this issue by using multiresolution temporal patterns as our self-supervision information. This motivation leads to our effective multiresolution self-supervised discrimination framework for time series anomaly detection.

It is worth noting that self-supervised technology is different from data augmentation. Data augmentation focus on sampling more informative samples to alleviate the shortage of limited data, while the self-supervised method focus on learning representation via auxiliary tasks. For example, in the field of time series anomaly detection, we use downsampling to obtain temporal information at different resolutions and then the self-supervised technology can be leveraged to describe normality by learning a discriminator to distinguish multiresolution information by pseudo-labels. After that, a test sample will be detected based on the discriminator's confidence to classify the sample under multiple temporal resolutions. If the discriminator's classification confidence is lower than a predefined threshold, it will identify the test sample as the anomalous. Compared to the self-supervision methods, traditional time series data augmentation techniques depend on operations including warping, adding noises. They do not provide informative self-supervised pseudo-labels to help the model's training.

3. The Proposed Method

In this paper, we study the time series anomaly detection tasks in the one-class setting [40–42]. This means that we train our model on pure normal data and only the testing dataset contains unknown anomalies. Our goal aims to identify abnormal time ser-

ies segments in testing dataset with a higher accuracy and a lower computational cost.

We show the proposed MS^{2}D-Net in Fig. 3. The MS^{2}D-Net mainly includes three components. The first part is a multiresolution downsampling module to create multiresolution temporal patterns with self-supervision pseudo-labels, the second part is a feature extraction module based on a shallow convolutional neural network, and the last part is a self-supervised discrimination module. For time series anomaly detection, a normalized anomalousness score can be calculated based on the self-supervised discrimination loss. In the following sections, we introduce each component formally.

3.1. Multiresolution Downsampling Module

For a time series, we can always describe it from different time resolutions, such as day, week, month, etc. Based on this observation, our multiresolution downsampling module directly extracts informative self-supervision information via a simple downsampling strategy from multiple time resolutions. Also, the corresponding downsampling rates will be used as pseudo-labels for training.

Given a time-series dataset $\mathscr{D} = \{\mathbf{X}_i\}_{i=1}^{N}$, N denotes the number of samples and $\mathbf{X}_i$ is defined as an M-dimensional T-length time series:

$$\begin{aligned}\mathbf{X}_i &= (\mathbf{X}_i(1), \cdots, \mathbf{X}_i(t), \cdots, \mathbf{X}_i(T)) \\ &= \begin{pmatrix} x_{1,1} & \cdots & x_{1,t} & \cdots & x_{1,T} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,t} & \cdots & x_{m,T} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{M,1} & \cdots & x_{M,t} & \cdots & x_{M,T} \end{pmatrix},\end{aligned} \tag{1}$$

where $\mathbf{X}_i(t) \in \mathbb{R}^{M\times 1}$ and $x_{m,t}$ denotes the observed values at the t-th timastamp in the m-th dimension.

For each sample $\mathbf{X}_i$, we can generate a series of augmented samples based on multiple predefined downsampling rates [43]. Generated sample can present abundant dynamic behaviors of the original time series at multiple time resolutions. Let the downsampling rate be denoted as f and $f \in \{1, 2, \cdots, K-1, K\}$,where K is the number of downsampling rates we used. Then, a downsampled time series with the downsampling rate f can be formulated as:

$$\mathbf{X}_i^f = (\mathbf{X}_i(1), \mathbf{X}_i(f+1), \cdots, \mathbf{X}_i(l*f+1)), \tag{2}$$

where $l = \lfloor (T-1)/f \rfloor$. Then, we can create a pseudo-label f for the downsampled time series $\mathbf{X}_i^f$. Based on these pseudo-labels, all downsampled samples can be assumed from a distribution with categories $1, 2, \ldots, K$. Then, the original anomaly detection problem can be cast into a supervised discrimination problem. Since our multiresolution self-supervised discrimination framework is based on multiresolution downsampling, the obtained downsampled instances $\{\mathbf{X}_i^f\}$ are with various lengths. In order to facilitate implementation, we pad zeros at the end of samples of unequal length such that all data are of equal length as shown in Fig. 4. Note that after zero-padding, although all instances are of equal length, they still show different temporal characteristics from various time scales.

3.2. Feature Extraction Module

Convolution neural network has shown its superiority on feature learning. Our feature extraction module is based on a shallow convolution neural network. As shown in Fig. 3, we use two convolution blocks in our feature extraction module and each of them consists of a convolutional layer and a max-pooling layer.

Formally, let convolutional filters be denoted by $\mathbf{W}_b \in \mathbb{R}^{l\times N_{in}\times N_{out}}$ where b denotes the b-th convolution block ($b \in \{1, \ldots, B\}$, B is the number of convolution blocks (B is set to be 2 in this work), l denotes the length of convolutional filters on time direction, N_{in} denotes the channels of the input and N_{out} denotes the output channels of convolutional layer. Moreover, we denote the input of the b-th block as $\mathbf{Z}_b$. When b equals to one, $\mathbf{Z}_b$ will be a time series $\mathbf{X}$ in the original input space. Here, we omit the subscript i for simplicity.

For the b-th convolution block, the convolutional result of input $\mathbf{Z}_b$ is computed by:

$$\mathbf{F}_b = g(\mathbf{W}_b * \mathbf{Z}_b + \mathbf{B}_b), \tag{3}$$

where $\mathbf{F}_b$ denote the convolution outputs. $*$ denotes the convolution operator. g is a nonlinear activation function (e.g., ReLU). The term

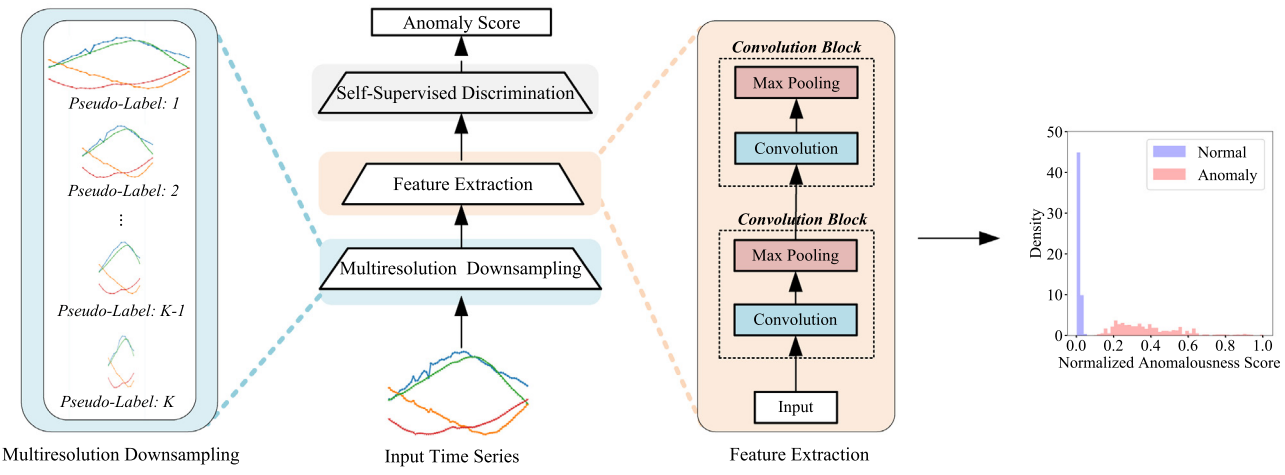


Fig. 3. The Architecture of the Proposed MS^{2}D-Net. Specifically, the MS^{2}D-Net consists of three parts, including a Multiresolution Downsampling Module, a Feature Extraction Module and a Self-Supervised Discrimination Module. Based on the loss of the Self-Supervised Discrimination Module (a linear classifier), a Normalized Anomalousness Score can be defined for efficient time series anomaly detection.

$\mathbf{B}_b$ is the corresponding bias. Subsequently, a max-pooling layer is applied on each channel of $\mathbf{F}_b$ as

$$\mathbf{P}_b = \text{MaxPooling}(\mathbf{F}_b), \tag{4}$$

where $\mathbf{F}_b$ and $\mathbf{P}_b$ denote the b-th convolution block's features and pooling features, respectively. In the subsequent $(b+1)$-th convolution block, the features $\mathbf{P}_b$ will be used as its input $\mathbf{Z}_{b+1}$.

In a summary, if $\mathscr{F}$ and $\mathscr{B}_b$ denote our feature extractor and b-th convolution block respectively, then we can formulate the feature extractor's output by

$$\mathbf{Z}(\boldsymbol{\theta}) = \mathscr{F}(\mathbf{X};\boldsymbol{\theta}) = \mathscr{B}_B \circ \mathscr{B}_{B-1} \circ \ldots \circ \mathscr{B}_1(\mathbf{X}), \tag{5}$$

where $\mathscr{B}_b$ is the concatenation of the convolution operation (Eq. 3) and and the max-pooling operation (Eq. 4). All parameters in the network are absorbed into $\boldsymbol{\theta}$ including convolution filters and biases.

3.3. Self-Supervised Discrimination Module

For each sample $\mathbf{X}_i$ (i denotes the index of sample), we can collect samples $\{\mathbf{X}_i^f\}$ generated by multiresolution downsampling, where f denotes the index of the f-th downsampling rate and $f \in \{1, 2, \cdots, K-1, K\}$. Each $\mathbf{X}_i^f$'s one-hot label is denoted as $\mathbf{y}_i^f \in \mathbb{R}^K$. Its c-element $\mathbf{y}_{i,c}^f$ is given by

$$\mathbf{y}_{i,c}^f = \begin{cases} 0, & c \neq f, \\ 1, & c = f. \end{cases} \tag{6}$$

Based on a linear mapping parameterized by $\mathbf{W}^c$ (with a shape of K-by-Q, Q is the output dimension of feature extraction module), a softmax layer can be used to define the probability of a downsampled time series $\mathbf{X}_i^f$ belonging to k-th pseudo-label category. Formally, this probability is formulated by

$$\begin{aligned} P(\mathbf{y}_{i,c}^f = k|\mathbf{X}_i^f;\boldsymbol{\theta}) &= \text{Softmax}(\mathbf{Z}(\boldsymbol{\theta})) \\ &= \frac{\exp(\mathbf{W}^k\mathbf{Z}(\boldsymbol{\theta}))}{\sum_{c=1}^{K}\exp(\mathbf{W}^c\mathbf{Z}(\boldsymbol{\theta}))}, \end{aligned} \tag{7}$$

where $\mathbf{Z}(\boldsymbol{\theta})$ is a Q-dimension vector generated from feature extraction module.

Then, we can define a self-supervised discrimination loss for sample $\mathbf{X}_i$ as $\mathscr{L}_{SS}(\mathbf{X}_i;\boldsymbol{\theta})$. Specifically, it is defined by the average cross-entropy loss of its all downsampled instance $\{\mathbf{X}_i^f\}$. That is

$$\begin{aligned} \mathscr{L}_{SS}(\mathbf{X}_i|\boldsymbol{\theta}) &= -\tfrac{1}{K}\sum_{f=1}^{K}\sum_{c=1}^{K}\mathbf{y}_{i,c}^f \log\left(P(\mathbf{y}_{i,c}^f = k|\mathbf{X}_i^f;\boldsymbol{\theta})\right) \\ &= -\tfrac{1}{K}\sum_{f=1}^{K}\log\left(P(\mathbf{y}_{i,c}^f = f|\mathbf{X}_i^f;\boldsymbol{\theta})\right). \end{aligned} \tag{8}$$

Finally, we can get the overall objective as

$$\mathscr{L} = \min_{\boldsymbol{\theta}}\frac{1}{N}\sum_{i=1}^{N}\mathscr{L}_{SS}(\mathbf{X}_i|\boldsymbol{\theta}), \tag{9}$$

As can be seen in our objective, the proposed method is based on cross-entropy losses over multiple downsampled instances that are with K pseudo labels corresponding to K time resolutions.

Next, we will formally give the anomaly score formulation to clarify how to use the trained multiresolution self-supervised discrimination module for time series anomaly detection.

3.4. Anomaly Score (AS)

Given a test sample $\mathbf{X}_{\text{test}}$, we downsample it with K different downsampling rates and get augmented time series $\{\mathbf{X}^f\}$. Then its anomaly score value can be given by

$$\text{AS}(\mathbf{X}_{\text{test}}) = -\frac{1}{K}\sum_{f=1}^{K}\log\left(P(\mathbf{y}_{\text{test},c}^f = f|\mathbf{X}_{\text{test}}^f;\boldsymbol{\theta})\right). \tag{10}$$

To make an anomaly score range from [0, 1], we further introduce the normalized version of the anomaly score denoted as $(\tilde{\text{AS}})$:

$$\tilde{\text{AS}}(\mathbf{X}_{\text{test}}) = \frac{\text{AS}(\mathbf{X}_{\text{test}}) - \min(\text{AS}(\mathbf{X}_{\text{test}}))}{\max(\text{AS}(\mathbf{X}_{\text{test}})) - \min(\text{AS}(\mathbf{X}_{\text{test}}))}. \tag{11}$$

So far, our time series anomaly detection framework based on the self-supervised discriminative network has been established. We provide a detailed algorithm flow of the proposed MS^{2}D-Net in Algorithm 1.

Algorithm 1 The proposed MS^{2}D-Net

Input: Time series set; number of different downsampling rates K; maximum number of iterations max_Iter; learning rate α.
Training phase:
for each data instance $\mathbf{X}_i$ **do**
  for $f \in [1, 2, \ldots, K]$ **do**
    Downsample time series $\mathbf{X}_i^f$ with the donwsampling rate f by Eq. (2).
    Create its pseudo-label $\mathbf{y}_i^f$.
  end for
end for
$\boldsymbol{\theta} \leftarrow$ Initialize network parameters.
while iter < max_Iter **do**
  Obtain the extracted feature $Z(\boldsymbol{\theta})$ by Eq. (3–5)
  Compute the self-supervised discrimination loss L by Eq. (7–9).
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha\nabla_{\theta}L$.
  iter+ = 1.
end while
return Trained network.
Testing phase:
Given a test sample $\mathbf{X}$
for $f \in [1, 2, \ldots, K]$ **do**
  Downsample time series $\mathbf{X}^f$ with the donwsampling rate f by Eq. (2).
  Create its pseudo-label $\mathbf{y}^f$.
end for
$\boldsymbol{\theta} \leftarrow$ Load the trained network's weights.
Obtain its extracted feature $Z(\boldsymbol{\theta})$ by Eq. (3–5)
Calculate its anomaly score by Eq. (10).
return Anomaly score.

4. Experiments

To evaluate the performance of MS^{2}D-Net, we conduct experiments on 18 time series anomaly detection tasks. In this section, we will firstly describe our experimental setup. Then, we will introduce the baselines used in our experiments. Subsequently, detailed analysis on visualization and hyperparameter sensitivity will be provided. Finally, the analysis on the computational efficiency of our model will be further discussed.

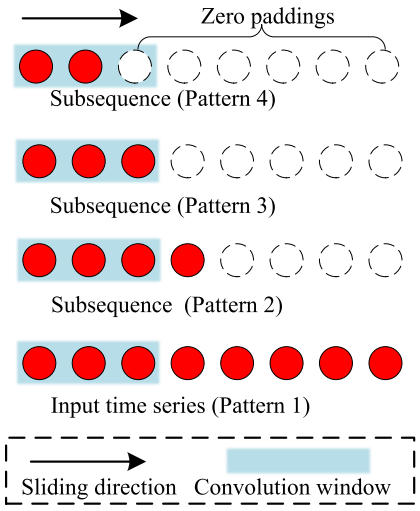


Fig. 4. Illustration of convolution-based feature extraction for multiresolution downsampling subsequences.

Table 1
Summary of different datasets. Anomaly Ratio shown in the last column corresponds to the testing set.

Dataset	#Train	#Test	Anomaly Ratio
Mocap	980	749	0.81
NYC Taxi	747	426	0.06
2D Gesture	833	285	0.31
Power Demand	1,248	1,459	0.13
nprs43	1,248	722	0.21
nprs44	1,248	1,093	0.12
TEK14	833	184	0.49
TEK16	833	80	0.33
TEK17	833	77	0.27
chf01	833	169	0.25
chf13	833	113	0.28
chfdbchf15	833	319	0.10
ltstdb43	833	97	0.29
ltstdb240	833	129	0.23
mitdb180	833	210	0.17
qtdbsel102	833	973	0.04
stdb308	833	257	0.16
xmitdb108	833	160	0.34

4.1. Experimental Setup

4.1.1. Datasets

Table 1 summarizes the basic statistics of selected datasets (after preprocessing). A brief description for our datasets is given as follows:

- Motion Capture dataset[2] contains different motions performed by multiple subjects, including walking, climbing, running, etc.
- NYC Taxi Passenger Count dataset [44] collects the New York City taxi passenger data stream, which is aggregated at 30-min intervals.
- 2D Gesture dataset [16] contains X-Y coordinates of hand gestures in a video along time.
- Power Demand dataset [16] is a real-world power demand dataset that measured the power consumption for a Dutch research facility for the entire year of 1997.
- Respiration [16] includes two datasets (nprs43 and nprs44), each of which collects the patient's respiration measured by thorax extension, and the sampling rate is 10 Hz.
- Space Shuttle [16] includes three datasets (TEK14, TEK16, and TEK17), each of which contains Space Shuttle Marotta Valve time series that was annotated by a NASA engineer.
- Electrocardiograms (ECGs) [16] includes 9 real-world datasets (chf01, chf13, chfdbchf15, ltstdb43, ltstdb240, mitdb180, qtdbsel102, stdb308, and xmitdb108), each of which contains a single anomaly subsequence corresponding to a pre-ventricular contraction.

4.1.2. Implementation Details

Before training, we follow the method recommended by BeatGAN [22] to preprocess the Mocap dataset. For the other 17 datasets, we use the preprocessing methods from [45].

For each dataset, we use a sliding window to extract time series segments. Specifically, for the Power Demand and Respiration dataset, we use a window of size 200 with a sliding step of 80 on the training datasets. For other datasets, we use a window of 160 with a sliding step of 120 (on the training datasets). For the testing datasets, the sliding step is set to 10 so that the size of the testing dataset can be kept in a decent size. A sample is considered to be anomalous if there exist any anomaly timestamps within the segment.

For the proposed MS^{2}D-Net, we set the number of the downsampling rates K as a hyperparameter searched from the set {10, 20, 30, 40, 50}. In the feature extraction module, we use a shallow convolutional neural network consisting of two convolution blocks (including max-pooling). For the first convolution layer, its input channel (N_{in}) is set to be the input time series dimension (M) and its output channel is set as 16. For the second convolution layer, the input channel is set as 16 and the output channel is set as 32. The convolutional filter length l (on the time direction) is searched from the set {1, 3, 5}. To allow gradient training, we use the Adam optimizer [46] with an initial learning rate $\alpha = 0.001$, and the weight decay rate of 0.0001. All the experiments are run on an Intel Core CPU E5 – 2620 v4 @ 2.10GHz CPU, 64 GB RAM and a GeForce GTX 1080-Ti 11G GPU.

Note that although traditional anomaly detection methods such as OCSVM and LOF usually focus on static features (rather than temporal data), we still can treat the time series as a high-dimensional feature vector by ignoring temporal dependencies. For this reason, we also compare these traditional yet popular machine learning baselines with the proposed model. For the fairness of comparisons, we use the same sliding windows to transform time series into to corresponding static feature vectors.

4.1.3. Evaluation Metrics

For evaluation, we employ two commonly used metrics, Area under the Curve of Receiver Operating Characteristic (AUROC) [47] and Area under the Curve of Precision-Recall (AUPR) [47]. The AUROC reflects the trade-off among true positives, true negatives, false positives, and false negatives. We also use the AUPR metrics, which are the case of using outliers as the positive class. Higher AUROC and AUPR values indicate better performance. Both of them are popular metrics for anomaly detection tasks [36].

Intuitively, AUROC indicates the correctness of the model for predicting the permutation between positive samples and negative samples where positive samples and negative samples correspond to the anomalous samples and normal samples respectively. When the predicted permutation of a positive sample is ranking before a negative one, it is considered to be a good model inference. Consider we has m positive samples and n negative samples in the testing stage while the anomaly score is AS, AUROC metric can be calculated as

[2] http://mocap.cs.cmu.edu/

$$\text{AUROC} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}\mathbf{I}(AS(i) > AS(j))}{m * n}, \tag{12}$$

where $\mathbf{I}(\cdot)$ is the indicator function and equals to 1 if the condition is true, veras equals to 0. i, j denotes the index of normal samples and abnormal samples in the testing stage, respectively.

The AUPR metric corresponds to the area under the curve, constructed by the precision and recall of positive sample for a method. The AUPR metrics generally include two specific indicators: the AUPR-in and AUPR-out, which indicates the normal samples or the abnormal one as the positive sample. Importantly, AUPR-out (AUPR shown in this paper) is more excellent to evaluate different anomaly detection methods since a random guess gets an anomaly ratio for the AUPR metric (which is small in most situations), while a random guess gets a 0.5 score for the AUROC metric. In this paper, we also present the AUPR-in result in the appendix.

4.1.4. Baselines

We compare the proposed MS^{2}D-Net with several popular anomaly detection methods, including tradictional machine learning methods: One-class Support Vector Machines (OCSVM) [27], Local Outlier Factor (LOF) [25], Isolation Forest (ISF) [23], Matrix Profile (MP) [24], and recent strong deep learning baselines LSTM-based predictor (LSTM_predictor) [5], Sequence2Sequence (Seq2Seq) [10], BeatGAN [22] and THOC [48]. We briefly introduce them as follows:

- One-class SVM (OCSVM) is a popular variant of support vector machines (SVMs) for anomaly detection, which learns a discriminative hyperplane to describe normal data [27].
- Local outlier factor (LOF) [25] measures the degree of abnormality based on the mean value of the ratio of the local reachability density of a sample and those of its k-nearest neighbors.
- Isolation Forest (ISF) builds an ensemble of decision trees, then the anomalies will be detected if they have short average path lengths on the learned decision trees [23].
- Matrix Profile (MP) calculates the Euclidean distance between the query time series segment and the most similar subsequence segment. This distance is used as an indicator of anomaly detection [24]. The larger the distance, the more likely it is an anomaly. It is one of the most competitive time-series anomaly detection methods.
- LSTM_predictor (LSTM_pre) [5] predicts the value at the next time step based on its historical observations. Then the prediction error can be used as an anomaly score to perform anomaly detection on inference.
- Seq2Seq model [10] reconstructs the time series segments in a reverse time order as the training objective of the network. The reconstruction error is directly used as the corresponding anomaly score.
- BeatGAN [22] incorporates a convolutional auto-encoder network incorporated with an adversarial regularization for reconstruction. Also, the reconstruction error is used to define the anomaly score.
- Temporal Hierarchical One-Class Network (THOC) [48] is a recent strong baseline for time series anomaly detection. It extends traditional single-ball based one-class model by using multiple hierarchically structured hyperspheres.

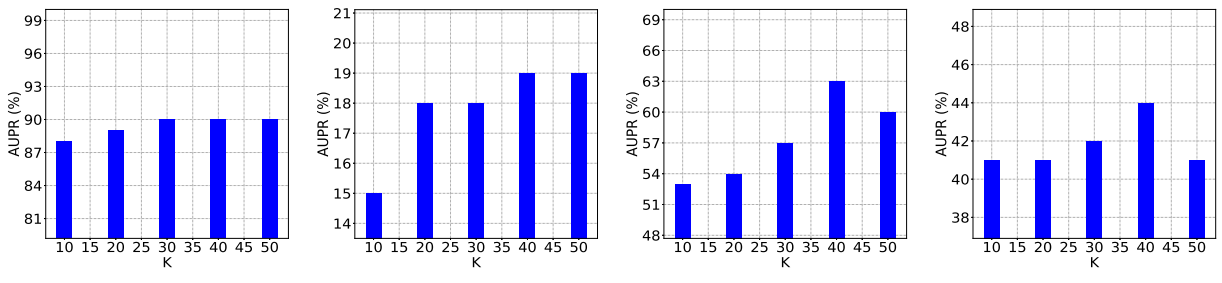
4.2. Hyperparameter Sensitivity

In our self-supervised discriminative framework, the number of pseudo-labels (or categories) K is a crucial hyperparameter. Thus, we further provide a detailed analysis on this hyperparameter. Specifically, we test its effects on four datasets, varying its values in the range of $\{10, 20, 30, 40, 50\}$. Results in terms of the AUPR and the AUROC are reported in Fig. 5 and Fig. 6 respectively. We can see that MS^{2}D-Net prefers a larger K. When K is too small, the model performance is degraded due to insufficient time-scale information. When K is much larger, the model may suffer from redundant patterns, which is not discriminative enough and may hamper the model's training.

4.3. Results and Anylasis

Results in terms of the AUPR and AUROC on 18 time series anomaly detection datasets are reported in Table 2 and Table 3, respectively. The best result is highlighted in bold. Also, we report the average results and the average rank of each method. As can be seen, our MS^{2}D-Net achieves the best performance on 8 out of 18 tasks and achieves the highest rank among all of the baselines, which demonstrates its effectiveness. According to the results, we can further draw the following conclusions:

a) Compared to traditional machine learning anomaly detection methods, MS^{2}D-Net outperforms them in terms of average accuracy. Although MP and LOF perform better (with higher average rank) than OCSVM and ISF, they still easily suffer from two problems. The first one is that they need to traverse all training samples during testing. The other limitation is that MP is a method based on distance alignment, which is limited to time series distortions. Our MS^{2}D-Net can inference the test set at one time and smooth the local time series distortions by the multiresolution downsampling module.
b) MS^{2}D-Net is better than existing deep learning-based baselines. Compared to the baseline LSTM_predictor, MS^{2}D-Net can improve by around seven percentages on AUPR and improve around one percentage on AUROC. The regression capacity of the LSTM_predictor can hardly model the normal samples for distinguishing the normal patterns, while the MS^{2}D-Net can recognize the normal patterns in a multiresolu-



(a) 2D Gesture (b) Power Demand (c) nprs43 (d) nprs44

Fig. 5. The AUPR indicator among different self-supervised categories K on four datasets.

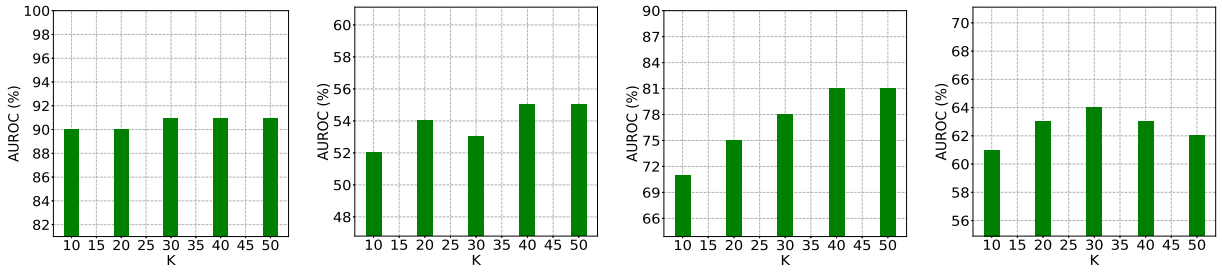


Fig. 6. The AUROC indicator among different self-supervised categories K on four datasets.

Table 2
Comparison performance in terms of AUPR indicator on 18 time series datasets.

Dataset	OCSVM	LOF	ISF	MP	BeatGAN	Seq2Seq	LSTM_pre	THOC	Ours
Motion Capture	0.8853	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.8439	0.9833	**1.0000**	**1.0000**
NYC Taxi	0.1027	0.1152	0.0903	0.1152	0.1182	0.0600	**0.1954**	0.1951	0.0737
2D Gesture	0.6956	0.8924	0.5839	0.5828	0.8009	0.6898	0.6531	0.8840	**0.9020**
Power Demand	0.1165	0.1744	0.1018	0.1065	0.1049	**0.3323**	0.2430	0.2700	0.1873
nprs43	0.5683	0.5691	0.5465	0.5691	0.5864	0.5989	0.5777	0.6205	**0.6279**
nprs44	0.2817	0.4169	0.1655	0.2509	0.4064	0.4172	0.4109	0.3498	**0.4374**
TEK14	0.4152	0.5688	0.5924	**0.6737**	0.4981	0.6236	0.4696	0.6062	0.6272
TEK16	0.7458	0.7592	0.7810	0.7051	0.7217	**0.8782**	0.5963	0.5837	0.8007
TEK17	0.3267	0.2258	0.5336	0.2258	0.3216	0.3413	0.4927	**0.5351**	0.4623
chf01	0.7833	0.9105	0.7077	0.9000	0.7666	0.7157	0.7730	**0.9726**	0.9151
chf13	0.8503	0.9174	0.7956	0.8231	0.8355	0.7910	0.8804	0.9092	**0.9364**
chfdbchf15	0.6149	0.8744	0.4829	0.8471	0.6828	0.2856	0.7277	0.8936	**0.9532**
ltstdb43	0.3813	0.4331	0.4289	0.4121	0.3324	0.4819	0.4938	**0.5976**	0.4686
ltstdb240	0.9016	0.8986	0.7266	0.8520	**0.9258**	0.9153	0.9230	0.5578	0.8434
mitdb180	0.2731	0.2164	0.1746	0.2569	0.2949	0.3831	0.2438	0.1537	**0.6265**
qtdbsel102	0.5596	0.7643	0.1123	0.5478	0.2308	0.1268	0.5776	**0.8215**	0.8064
stdb308	0.1940	0.1465	0.2560	0.1385	0.2653	0.2713	**0.3596**	0.2049	0.1419
xmitdb108	0.2622	0.6137	0.2516	0.5363	0.2750	0.3838	0.5348	0.6111	**0.8727**
No. best	0	1	1	2	2	2	2	5	**8**
Avg	0.5115	0.5814	0.4753	0.5298	0.5231	0.5151	0.5648	0.5974	**0.6359**
AVG. Rank	6.2222	4.4444	6.9160	5.9444	5.3611	4.9444	4.5000	3.6944	**2.9722**

[1] The "No. best" refers to the number of tasks for which a method achieves the best result in experimental tasks.

Table 3
Comparison performance in terms of AUROC indicator on 18 time series datasets.

Dataset	OCSVM	LOF	ISF	MP	BeatGAN	Seq2Seq	LSTM_pre	THOC	Ours
Motion Capture	0.5559	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.6120	0.9959	**1.0000**	**1.0000**
NYC Taxi	0.5830	0.5680	0.5415	0.5680	0.5369	0.5140	**0.6197**	0.5664	0.5106
2D Gesture	0.8086	0.9089	0.7874	0.8286	0.8691	0.7143	0.7941	0.8906	**0.9130**
Power Demand	0.4794	0.5065	0.3977	0.3995	0.4202	**0.7572**	0.7093	0.6297	0.5393
nprs43	0.7617	0.7516	0.7557	0.7516	0.7643	0.7966	0.7689	0.8063	**0.8081**
nprs44	0.5379	0.6153	0.5117	0.5712	0.5642	0.6127	0.6222	0.5895	**0.6240**
TEK14	0.3827	0.6936	0.4548	0.6142	0.5352	0.6849	0.4690	0.5773	**0.7048**
TEK16	0.8490	0.8084	0.8981	0.8397	0.8846	**0.9430**	0.8105	0.8504	0.7756
TEK17	0.5510	0.3895	**0.7554**	0.3895	0.5374	0.5799	0.7287	0.7194	0.4582
chf01	0.7859	0.9454	0.7436	0.9265	0.7743	0.8455	0.7962	**0.9886**	0.9433
chf13	0.8399	0.9429	0.8110	0.8287	0.8349	0.8511	0.9336	0.9110	**0.9582**
chfdbchf15	0.7735	0.9140	0.7384	0.9091	0.8550	0.7109	0.8883	0.9538	**0.9801**
ltstdb43	0.4415	0.6201	0.5581	0.5455	0.4130	0.6325	0.6056	**0.6762**	0.6056
ltstdb240	0.9721	0.9620	0.9156	0.8552	**0.9727**	0.9680	0.9717	0.8513	0.9272
mitdb180	0.7032	0.6144	0.5629	0.7094	0.7509	**0.8031**	0.6838	0.4226	0.7240
qtdbsel102	0.8069	0.9134	0.6283	0.7986	0.6801	0.7319	0.8850	**0.9447**	0.9137
stdb308	0.6150	0.4757	0.6553	0.4277	0.6893	0.6677	**0.7013**	0.6106	0.4577
xmitdb108	0.3445	0.7116	0.3303	0.6144	0.3966	0.4602	0.5629	0.6000	**0.8870**
No. best	0	1	2	1	2	3	2	4	**8**
Avg	0.6551	0.7412	0.6767	0.6987	0.6933	0.7159	0.7526	0.7549	**0.7628**
AVG. Rank	6.0556	4.3333	6.8055	5.8889	5.4720	4.6111	4.2500	3.9167	**3.6667**

[1] The "No. best" refers to the number of tasks for which a method achieves the best result in experimental tasks.

tion way and thus make good performances for distinguishing the abnormal patterns from the normal ones.
c) Compared to the baselines Seq2Seq and BeatGAN, MS^{2}D-Net still shows great improvements. The Seq2Seq is easily limited by the regression capacity of the recurrent neural network while the BeatGAN suffers from unstable adversarial training. Different from these two methods, our MS^{2}D-Net is a simple yet efficient self-supervised discrimination framework. It doesn't depend on regression modeling or adversarial training. By learning multiresolution self-supervision information from normal data, normality can be described effectively.
d) Compared to the recent strong baseline THOC, our MS^{2}D-Net still show improvements by around four percentages on AUPR and around one percent on the AUROC indicator. This means that the clustering space of the THOC is also hard to distinguish the abnormal patterns from the normal ones. Contrarily, the multiresolution downsampling space in MS^{2}D-Net is more proper for modeling normal samples, which helps the model distinguish the abnormal patterns from the normal ones.

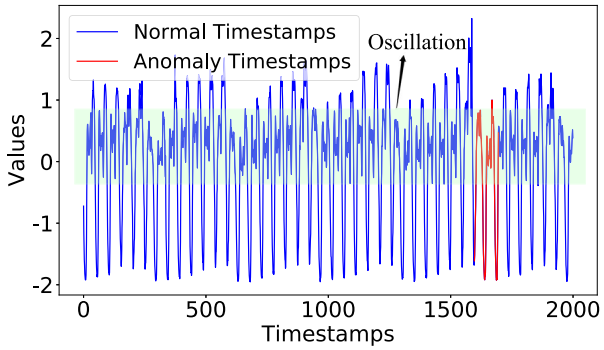


Fig. 7. Visualization of the NYC Taxi (Traffic flow after normalization). The red curve corresponds to abnormal subsequence. Due to the existence of the high-frequency oscillation (green zone), our model fails to detect anomalies.

Limitation and Discussion: Although the proposed MS^{2}D-Net shows improvements over 18 time series anomaly detection tasks, we observed that it still not good in some tasks (e.g., NYC Taxi). As can be seen in Fig. 7, there are a lot of highly nonlinear oscillations in NYC Taxi time series. Due to the extreme oscillation in time series, the multiresolution patterns may be not discriminative enough for describing the normal time series. This issue also hampers other prototype-based traditional machine learning methods since the shape of time series is important in their definition of anomaly score. Additionally, other deep learning baselines also suffer from the same problem and show poor performance on this dataset. Power Demand, TEK17 and stdb308 datasets also face similar problems. The characteristics of the samples in time series are not prominent, so the results of the model based on the subsequences are worse. This also encourages us to improve the proposed MS^{2}D-Net in more challenging dataset such as the NYC Taxi time series in our future works.

4.4. Visualization Analysis

To further understand the superiority of MS^{2}D-Net, we visualize the distribution of the normalized anomaly score on the Motion Capture dataset among LSTM_predictor, Seq2Seq, BeatGAN, THOC and our MS^{2}D-Net. Visualization results are shown in Fig. 8. For the Seq2Seq (Fig. 8(b)), there exists a large overlap between the score distributions of normal samples and anomalies. LSTM_predictor (Fig. 8(a)) is better than the Seq2Seq but there is still a small overlap area. Although both BeatGAN (Fig. 8)) and THOC (Fig. 8(d)) correctly identify the anomalies, our model's classification boundary is much more clear. This demonstrates that our model shows good robustness on time series anomaly detection tasks.

4.5. Computational Efficiency Analysis

The proposed MS^{2}D-Net is computationally efficient and has a faster inference time. This should be very beneficial in real

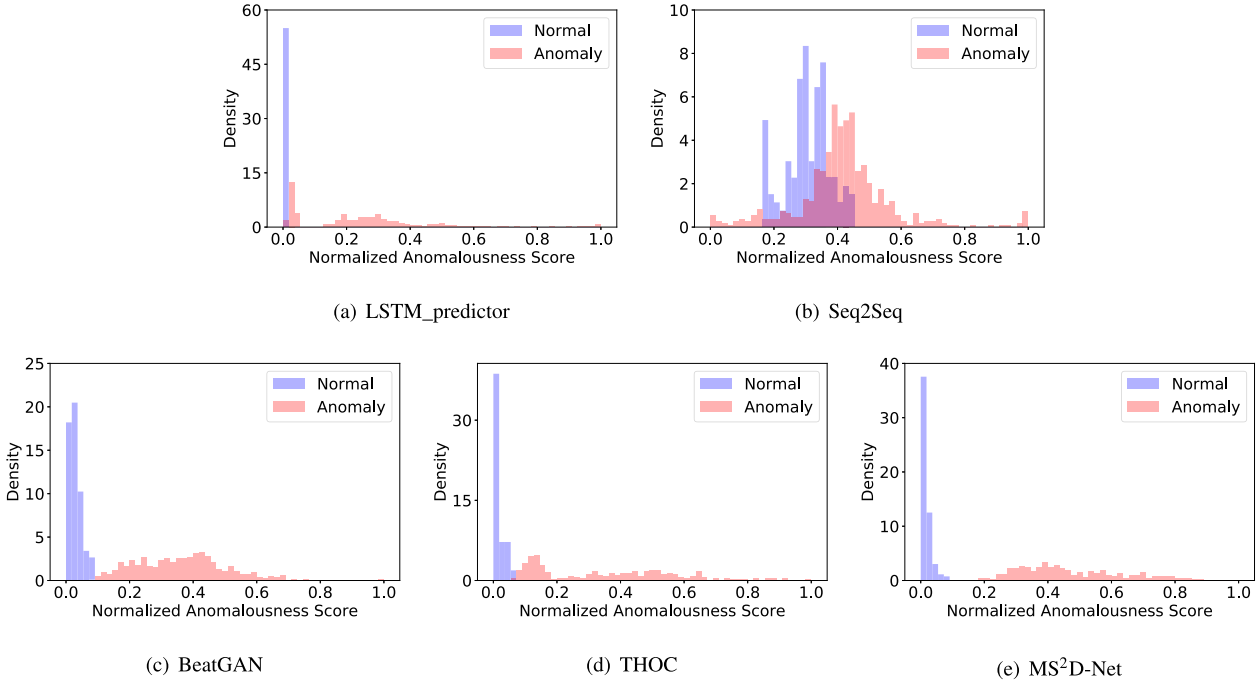
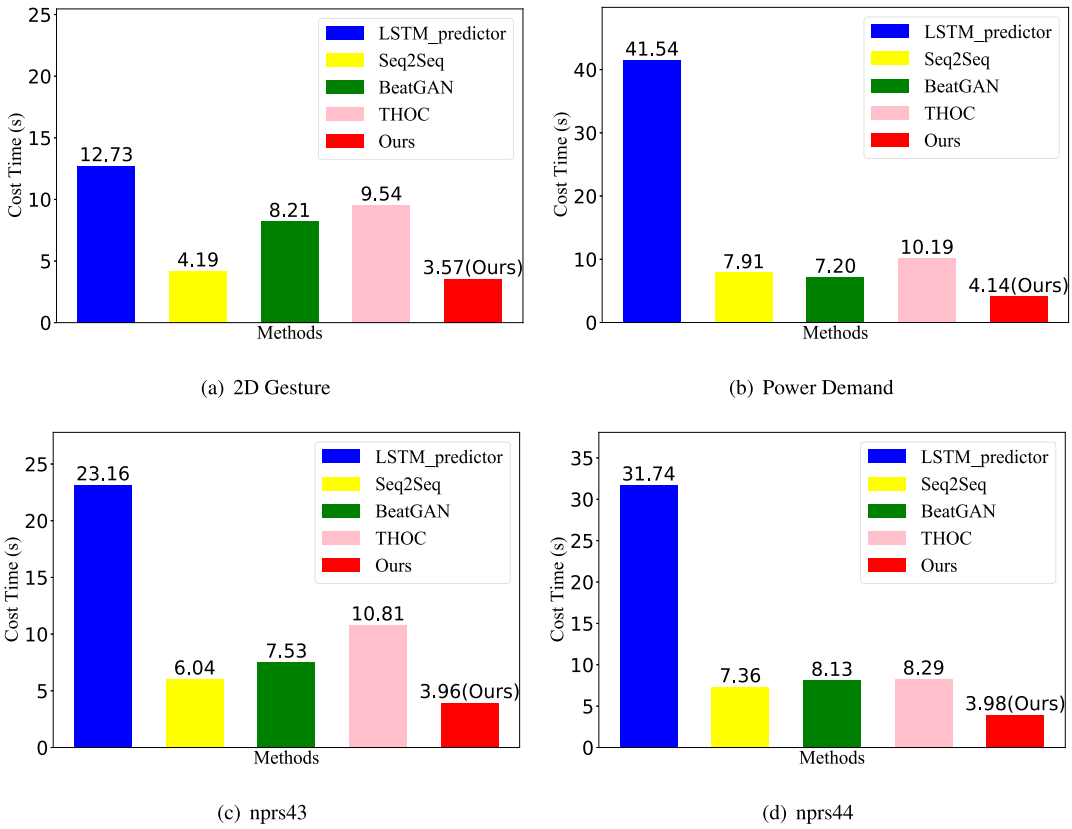


Fig. 8. Normalized Anomalousness Score Distributions on Mocap Dataset among deep learning methods.



(a) 2D Gesture (b) Power Demand

(c) nprs43 (d) nprs44

Fig. 9. Computational cost time among deep learning models and the proposed MS^{2}D-Net on four testing datasets.

application scenarios. In Fig. 9, we calculate the inference time at the testing stage on four datasets. Four strong deep learning baselines and our model are tested in this experiment. As can be seen, the proposed model framework consistently shows a faster inference speed than other baselines, especially faster than the recurrent model LSTM_predictor. Note that LSTM_predictor has a high computational cost during training. This is because that its RNN structure depends on the recurrent updates and its decoding procedure cannot be parallelized. Contrarily, our proposed MS^{2}D-Net is a discriminative method, which can be easily accelerated in its convolution layers. This supports the superiority of our model in terms of inference time.

5. Conclusion and Future Work

In this paper, a multiresolution self-supervised discrimination network (MS^{2}D-Net) is developed, which can identify anomalous time-series subsequences with a much lower computational cost. Its high computationally efficiency is achieved by integrating a multiresolution downsampling strategy with self-supervised discriminative learning. Compared to the existing time series detection methods, the proposed model does not depend on reconstruction (or self-prediction). In experiments, we demonstrate its superiority on 18 time series anomaly detection tasks.

In the future, we will take care of the time series oscillation problem in our proposed MS^{2}D-Net. Meanwhile, we will apply our MS^{2}D-Net on other topics, including the case of the training set containing unknown anomalies or missing values.

CRediT authorship contribution statement

Desen Huang: Software, Writing - original draft. **Lifeng Shen:** Writing - review & editing, Investigation. **Zhongzhong Yu:** Software, Validation. **Zhenjing Zheng:** Writing - review & editing. **Min Huang:** Supervision. **Qianli Ma:** Conceptualization, Methodology, Supervision.

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.

Acknowledgment

The work described in this paper was partially funded by the National Natural Science Foundation of China (Grant No. 61872148), the Natural Science Foundation of Guangdong Province (Grant Nos. 2017A030313355, 2019A1515010768 and 2021A1515011496), the Guangzhou Science and Technology Planning Project (Grant Nos. 201704030051, and 201902010020), the Key R&D Program of Guangdong Province (No. 2018B010107002) and the Fundamental Research Funds for the Central Universities.

Appendix A

Here, we present the comparison performance in terms of AUPR-in indicator on 18 time series datasets in Table 4.

Table 4
Comparison performance in terms of AUPR-in indicator on 18 time series datasets.

Dataset	OCSVM	LOF	ISF	MP	BeatGAN	Seq2Seq	LSTM_predict	THOC	Ours
Motion Capture	0.1919	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.3474	0.9990	**1.0000**	**1.0000**
NYC Taxi	0.9502	0.9439	0.9419	0.9439	0.9432	0.9538	**0.9545**	0.9413	0.9411
2D Gesture	0.8806	0.9353	0.8978	0.9217	0.9230	0.7547	0.9131	0.9232	**0.9373**
Power Demand	0.8857	0.8693	0.8532	0.8138	0.8704	**0.9522**	0.8457	0.8617	0.8782
nprs43	0.9253	0.9182	0.9207	0.9182	0.9195	0.8356	**0.9276**	0.9023	0.9155
nprs44	0.8650	0.9040	0.8663	0.9052	0.8869	0.9148	**0.9202**	0.8542	0.9161
TEK14	0.4458	**0.7973**	0.4458	0.5747	0.6438	0.7216	0.5567	0.6635	0.7382
TEK16	0.9361	0.8745	0.9579	0.9348	0.9543	**0.9758**	0.8843	0.9443	0.7984
TEK17	0.7739	0.7543	**0.8763**	0.7543	0.7413	0.7297	0.8126	0.8265	0.7102
chf01	0.8276	0.9786	0.8146	0.9681	0.8284	0.9402	0.8457	0.9543	**0.9862**
chf13	0.8948	0.9727	0.8721	0.8882	0.8864	0.9298	0.9736	0.8932	**0.9810**
chfdbchf15	0.9362	0.9688	0.9473	0.9839	0.9570	0.9265	0.9424	0.9714	**0.9972**
ltstdb43	0.6890	0.7783	0.7599	0.7323	0.6850	0.8224	0.7803	**0.8336**	0.7926
ltstdb240	**0.9923**	0.9885	0.9756	0.9316	0.9919	0.9905	0.9916	0.9413	0.9756
mitdb180	0.9211	0.8573	0.8906	0.9333	0.9465	0.9542	0.9219	**0.9569**	0.9103
qtdbsel102	0.9730	0.9930	0.9735	0.9881	0.9791	0.9771	0.9934	0.9647	**0.9953**
stdb308	0.9005	0.8617	0.9019	0.8198	**0.9216**	0.8847	0.8743	0.9023	0.8503
xmitdb108	0.5889	0.8365	0.6169	0.7336	0.6374	0.6490	0.7355	0.8840	**0.9087**
No. best	1	2	2	1	2	2	3	3	**7**
Avg	0.8099	0.9018	0.8618	0.8747	0.8731	0.8478	0.8818	0.9010	0.9018
AVG. Rank	5.9167	4.5000	6.0278	5.5000	4.9722	4.8333	4.8333	4.638889	4.166667

The "No. best" refers to the number of tasks for which a method achieves the best result in experimental tasks.

References

[1] Alaiñe Iturria, Jacinto Carrasco, Santi Charramendieta, Angel Conde, Francisco Herrera, otsad: A package for online time-series anomaly detectors, Neurocomputing 374 (2020) 49–53.
[2] Wu. Jia, Weiru Zeng, Fei Yan, Hierarchical temporal memory method for time-series-based anomaly detection, Neurocomputing 273 (2018) 535–546.
[3] Run-Qing Chen, Guang-Hui Shi, Wan-Lei Zhao, Chang-Hui Liang, A joint model for it operation series prediction and anomaly detection, Neurocomputing 448 (2021) 130–139.
[4] Subutai Ahmad, Alexander Lavin, Scott Purdy, and Zuha Agha. Unsupervised real-time anomaly detection for streaming data. Neurocomputing, 262:134–147, 2017. Online Real-Time Learning Strategies for Data Streams.
[5] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, Tom Soderstrom, Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining ACM, 2018, pp. 387–395.
[6] Nikolay Laptev, Saeed Amizadeh, Ian Flint, Generic and scalable framework for automated time-series anomaly detection, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining ACM, August 10–13, 2015, Sydney, NSW, Australia, 2015, pp. 1939–1947.
[7] Vinod Nair, Ameya Raul, Shwetabh Khanduja, Vikas Bahirwani, S. Sundararajan Sellamanickam, Sathiya Keerthi, Steve Herbert, Sudheer Dhulipalla, Learning a hierarchical monitoring system for detecting and diagnosing service issues, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining ACM, August 10–13, 2015, Sydney, NSW, Australia, 2015, pp. 2029–2038.
[8] Daehyung Park, Yuuna Hoshi, Charles C Kemp, A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder, IEEE Robotics and Automation Letters 3 (3) (2018) 1544–1551.
[9] Daehyung Park, Hokeun Kim, Yuuna Hoshi, Zackory M. Erickson, Ariel Kapusta, Charles C. Kemp, A multimodal execution monitor with anomaly classification for robot-assisted feeding, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017 IEEE, September 24–28, 2017, Vancouver, BC, Canada, 2017, pp. 5406–5413.
[10] 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.
[11] Hu. Jilin, Bin Yang, Chenjuan Guo, Christian S Jensen, Risk-aware path selection with time-varying, uncertain travel costs: a time series approach, The VLDB Journal-The International Journal on Very Large Data Bases 27 (2) (2018) 179–200.
[12] Manish Gupta, Jing Gao, Charu C Aggarwal, Jiawei Han, Outlier detection for temporal data: A survey, IEEE Transactions on Knowledge and Data Engineering 26 (9) (2014) 2250–2267.
[13] Seif-Eddine Benkabou, Khalid Benabdeslem, Bruno Canitia, Unsupervised outlier detection for time series by entropy and dynamic time warping, Knowl. Inf. Syst. 54 (2) (2018) 463–486.
[14] Su. Ya, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, Dan Pei, Robust anomaly detection for multivariate time series through stochastic recurrent neural network, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019 ACM, August 4–8, 2019, Anchorage, AK, USA, 2019, pp. 2828–2837.
[15] Dan Li, Dacheng Chen, Lei Shi, Baihong Jin, Jonathan Goh, and See-Kiong Ng. MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks. CoRR, abs/1901.04997, 2019.
[16] Eamonn Keogh, Jessica Lin, and Ada Fu. HOT SAX: Finding the most unusual time series subsequence: Algorithms and applications. In Proc. of the 5th IEEE Int'l Conf. on Data Mining, pages 440–449. Citeseer, 2004.
[17] Tharindu Fernando, Simon Denman, David Ahmedt-Aristizabal, Sridha Sridharan, Kristin R. Laurens, Patrick J. Johnston, Clinton Fookes, Neural memory plasticity for medical anomaly detection, Neural Networks 127 (2020) 67–81.
[18] Mikel Canizo, Isaac Triguero, Angel Conde, Enrique Onieva, Multi-head CNN-RNN for multi-time series anomaly detection: An industrial case study, Neurocomputing 363 (2019) 246–260.
[19] Maoguo Gong, Yu. Huimin Zeng, Hao Li Xie, Zedong Tang, Local distinguishability aggrandizing network for human anomaly detection, Neural Networks 122 (2020) 364–373.
[20] Chunyong Yin, Sun Zhang, Jin Wang, and Neal N. Xiong. Anomaly detection based on convolutional recurrent autoencoder for iot time series. IEEE Transactions on Systems, Man, and Cybernetics: Systems, pages 1–11, 2020.
[21] Weiying Xie, Jie Lei, Baozhu Liu, Yunsong Li, Xiuping Jia, Spectral constraint adversarial autoencoders approach to feature representation in hyperspectral anomaly detection, Neural Networks 119 (2019) 222–234.
[22] Bin Zhou, Shenghua Liu, Bryan Hooi, Xueqi Cheng, and Jing Ye. BeatGAN: Anomalous rhythm detection using adversarially generated time series. In IJCAI, pages 4433–4439. AAAI Press, 2019.
[23] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In ICDM, pages 413–422. IEEE, 2008.
[24] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. Matrix profile I: All pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets. In ICDM, pages 1317–1322. IEEE, 2016.
[25] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. LOF: Identifying density-based local outliers. In ACM sigmod record, pages 93–104. ACM, 2000.
[26] Xiaoyi Gu, Leman Akoglu, and Alessandro Rinaldo. Statistical analysis of nearest neighbor methods for anomaly detection. In NIPS, pages 10921–10931, 2019.
[27] Bernhard Schölkopf, John C. Platt, John Shawe-Taylor, Alex J. Smola, Robert C. Williamson, Estimating the support of a high-dimensional distribution, Neural Computation 13 (7) (2001) 1443–1471.
[28] Jue Wang and Anoop Cherian. Gods: Generalized one-class discriminative subspaces for anomaly detection. In ICCV, pages 8201–8211, 2019.
[29] Longyuan Li, Junchi Yan, Haiyang Wang, Yaohui Jin, Anomaly detection of time series with smoothness-inducing sequential variational auto-encoder, IEEE Transactions on Neural Networks and Learning Systems 32 (3) (2021) 1177–1191.
[30] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, Puneet Agarwal, Long short term memory networks for anomaly detection in time series In 23rd European Symposium on Artificial Neural Networks, 22–24, Bruges, Belgium, April, 2015, p. 2015.
[31] Tung Kieu, Bin Yang, Chenjuan Guo, Christian S. Jensen, Outlier detection for time series with recurrent autoencoder ensembles, in: Sarit Kraus (Ed.), Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019 ijcai.org, August 10–16, 2019, Macao, China, 2019, pp. 2725–2732.
[32] Y. Yoo, U. Kim, J. Kim, Recurrent reconstructive network for sequential anomaly detection, IEEE Transactions on Cybernetics (2019) 1–12.

[33] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In ICLR, 2018.
[34] Rodrigo Santa Cruz, Basura Fernando, Anoop Cherian, Stephen Gould, Deeppermnet: Visual permutation learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3949–3957.
[35] Izhak Golan and Ran El-Yaniv. Deep anomaly detection using geometric transformations. In NIPS, pages 9758–9769, 2018.
[36] Siqi Wang, Yijie Zeng, Xinwang Liu, En Zhu, Jianping Yin, Chuanfu Xu, and Marius Kloft. Effective end-to-end unsupervised outlier detection via inlier priority of discriminative network. In NIPS, pages 5960–5973, 2019.
[37] Weiying Xie, Xin Zhang, Yunsong Li, Jie Lei, Jiaojiao Li, and Qian Du. Weakly supervised low-rank representation for hyperspectral anomaly detection. IEEE Transactions on Cybernetics, 2021.
[38] Tao Jiang, Weiying Xie, Yunsong Li, Jie Lei, Qian Du, Weakly supervised discriminative learning with spectral constrained generative adversarial network for hyperspectral anomaly detection, IEEE Transactions on Neural Networks and Learning Systems (2021).
[39] Weiying Xie, Jie Lei, Shuo Fang, Yunsong Li, Xiuping Jia, Mingsuo Li, Dual feature extraction network for hyperspectral image analysis, Pattern Recognition 118 (2021) 107992.
[40] David Martinus Johannes Tax. One-class classification: Concept learning in the absence of counter-examples. Ph.D. thesis, Delft University of Technology, 2002.
[41] 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, 2018.
[42] Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla. Anomaly detection using one-class neural networks. arXiv preprint arXiv:1802.06360, 2018.
[43] Minshuo Chen, Lin Yang, Mengdi Wang, and Tuo Zhao. Dimensionality reduction for stationary time series via stochastic nonconvex optimization. In NIPS, pages 3496–3506, 2018.
[44] Yuwei Cui, Chetan Surpur, Subutai Ahmad, and Jeff Hawkins. A comparative study of HTM and other neural network models for online sequence learning with streaming data. In IJCNN, pages 1530–1538. IEEE, 2016.
[45] Jinman Park. RNN based Time-series Anomaly Detector Model Implemented in Pytorch. URL:https://github.com/chickenbestlover/RNN-Time-series-Anomaly-Detection, 2018.
[46] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.
[47] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In ICML, pages 233–240. ACM, 2006.
[48] Lifeng Shen, Zhuocong Li, and James Kwok. Timeseries anomaly detection using temporal hierarchical one-class network. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6–12, 2020, virtual, 2020.

Desen Huang Received the master's degree in School of Computer Science & Engineering, the South China University of Technology in 2021. He is currently working as a machine learning and data mining engineer in Baidu. His current research interests include machine learning, deep learning, and time series modeling.

Lifeng Shen Received the bachelor's degree in the Department of Mathematics, School of Information Science and Technology, Jinan University in 2015 and the master's degree in School of Computer Science & Engineering, the South China University of Technology in 2018. He is currently pursuing a Ph.D. degree with the Artificial Intelligence Thrust Area, Information Hub at the Hong Kong University of Science and Technology (HKUST). His research interests are machine learning algorithms, deep learning, and their applications for time series.

Zhongzhong Yu Is currently pursuing the master's degree in computer science and engineering from the South China University of Technology, Guangzhou, China. His current research interests include machine learning, deep learning, and time series modeling.

Zhenjing Zheng Is currently pursuing the master's degree in computer science and engineering from the South China University of Technology, Guangzhou, China. His current research interests include machine learning, deep learning, and time series modeling.

Min Huang Is the vice president and an associate professor of Software School of South China University of technology. Her research interests include Internet of things, group intelligence perception, big data processing, Chinese information analysis and processing, etc. In recent years, she has presided over some researching projects such as Natural Science Foundation Project of Guangdong Province, scientific and technological research projects of Guangdong Province. As a major member, she participated in a number of natural science funds and scientific and technological research projects of the state. She has published more than 60 research papers. And she is also being editor and reviewer of many international conferences and journals.

Qianli Ma (M'17) received the Ph.D. degree in computer science from the South China University of Technology, Guangzhou, China, in 2008. He is a Professor with the School of Computer Science and Engineering, South China University of Technology. From 2016 to 2017, he was a Visiting Scholar with the University of California at San Diego, La Jolla, CA, USA. His current research interests include machine learning algorithms, data-mining methodologies, and time-series modeling and their applications.