ITC 1/53
Information Technology and Control
Vol. 53 / No. 1 / 2024
pp. 37-52
DOI 10.5755/j01.itc.53.1.34254

An Anomaly Detection Approach Based on Bidirectional Temporal Convolutional Network and Multi-Head Attention Mechanism

Received 2023/05/30 | Accepted after revision 2023/09/11

HOW TO CITE: Wang, R., Li, J. (2024). An Anomaly Detection Approach Based on Bidirectional Temporal Convolutional Network and Multi-Head Attention Mechanism. *Information Technology and Control*, 53(1), 37-52. https://doi.org/10.5755/j01.itc.53.1.34254

An Anomaly Detection Approach Based on Bidirectional Temporal Convolutional Network and Multi-Head Attention Mechanism

Rui Wang
Shanxi Polytechnic College, Taiyuan, 030006, China

Jiayao Li
School of Software, Shanxi Agricultural University, Taigu, 030801, China

Corresponding author: Rui Wang, 17696066288@163.com

Anomaly detection aims at detecting the data instances that deviate from the majority of data, and it is widely used in various fields for its ability to ensure the quality of the overall data. However, traditional anomaly detection methods face the problems such as low efficiency due to high data complexity and lack of data labels. At the same time, most methods only learn the forward features of time-series data, while lacking attention to the reverse features. For these disadvantages, this paper designs an anomaly detection approach called BiTCN-MHA based on the bidirectional temporal convolutional network (BiTCN) and multi-head attention (MHA) mechanism, which learns the features of anomalous data by capturing the forward and reverse temporal features in the time-series data, as well as solves the problems of feature information overload and neuron "death" by using MHA mechanism and ELU activation function, respectively, thereby quickly and accurately detecting anomalous data. Extensive experiments on six public datasets show that compared with eight state-of-the-arts, the proposed BiTCN-MHA method can improve the precision, recall, AUC and F1-Score by about 6.10%, 10.16%, 4.06% and 8.50%, respectively, especially having better detection performance on small time-series data.

KEYWORDS: Anomaly Detection, Bidirectional Temporal Convolutional Network, Multi-head Attention Mechanism, ELU Activation Function.

1. Introduction

As a major form of data, time-series data is widely found in financial transactions [40], traffic network monitoring [35], and other industries, and the effectively processing of time-series data can enhance the effect of data-based prediction and monitoring. However, some abnormal data are often existing in the collected time-series data due to equipment failure, environmental changes and other factors, and the existence of abnormal data seriously affects the related work based on data, thus, there is an urgent need for an effective means to detect the abnormal data, thereby guaranteeing fthe safety of time-series data. For this reason, the anomaly detection methods aiming to detect outliers from the data that deviate from majority of data have been a hot research topic, and it has become one of the important means to ensure data quality [19].

To accurately detect anomalous data, scholars have successively proposed anomaly detection methods, such as statistical distribution-based method HBOS [17], clustering-based method SOM [30], direct projection-based method Z-Glyph [7], etc. feature association-based method [3], [4], [5], etc. However, these methods cannot autonomously learn the features of original temporal data and require massive samples to obtain better detection results, which causes these methods losing their application value in many scenarios. The development of deep learning technology makes it widely applied in image recognition, speech recognition, traffic detection and other fields. In comparison with traditional methods or machine learning models, deep learning technologies can enhance the efficiency of anomaly detection through its own powerful learning ability, therefore, deep learning-based approaches have been widely and deeply researched [32]. Specifically, deep learning-based approaches are mainly composed of SE (self-encoder)-based approaches and GAN (generative adversarial network)-based approaches. SE-based approaches train the models to learn the concept of real target classes by minimizing the differences between original images and generated images, while GAN-based approaches achieve Nash equilibrium by using a game approach to obtain better detection results.

However, the vast majority of existing deep learning-based anomaly detection methods cannot handle time-series data well, and thus causing ineffective in anomaly detection. To address the anomaly detection in a more targeted manner for the time-series data, many anomaly detection models have been intensively studied, such as RNN (Recurrent Neural Networks) [36], LSTM (Long Short-Term Memory) [20] and GRU (Gated Recurrent Units) [12]. Although these models show an improvement in detection performance over CNNs and their variants, these models cannot handle common dynamic periodic or acyclic patterns well in complex environments, especially the recursive models like LSTM require large computations and train slow as well as cannot continuously and effectively model the long-term trends. In addition, some time-series data contain semantic features that are bidirectional, and the unidirectional models have some shortcomings in detection due to learning only one-way features.

Since TCN (Temporal Convolutional Neural Network) can obtain the temporal features of time-series data much better, this paper proposes an anomaly detection method based on the bidirectional TCN and MHA (Multi-Head Attention) mechanism. It first changes the unidirectional structure of TCN model to bidirectional structure of BiTCN model, thereby learning the bidirectional semantic features of time-series data better; And then, it uses the MHA mechanism to provide larger weights to the features that cause the data to be judged as anomalous data. In addition, the commonly used activation function of ReLU is changed to ELU to avoid neuron "death" problem. The contributions of this paper can be summarized as follows:

1. We change the conventional unidirectional TCN model into a bi-directional structure (BiTCN) to capture the forward and reverse semantic features of time-series data. Specifically, the forward TCN model structure is used to extract the forward semantic features of preprocessed time-series data, and then invert the forward semantic features and input them to the reverse TCN model structure for extracting reverse features, finally the forward and reverse features are fused as the final features to be learned.
2. We replace the activation function of ReLU in the traditional TCN model with the ELU to help BiTCN model solve the neuron "death" problem. Specifical-

ly, we apply the ELU instead of ReLU to avoid the gradient of "0" when semantic feature is negative, thus alleviating the neuron "death" problem.

3 We apply the MHA mechanism to set larger weights to the features that cause the data to be judged as anomalous, thus avoiding the problem of feature information overload during the learning process.

4 We conduct extensive experiments to test the BiTCN-MHA method, the results verify that compared with eight state-of-the-arts, the proposed BiTCN-MHA can accurately detect anomalies from time-series data as well as has higher stability.

The remainder can be organized as follows. Section 2 introduces the related works on anomaly detection. Section 3 describes the anomaly detection method called BiTCN-MHA based on bidirectional TCN and MHA mechanism. Section 4 presents the experimental results. Finally, we summarize the whole paper and briefly explain the future work in Section 5.

2. Related Works

This section briefly reviewed the related works of traditional anomaly detection approaches and deep learning-based approaches.

2.1. Traditional Anomaly Detection Approaches

The traditional anomaly detection approaches are mainly composed of rule-based approaches, statistical-based approaches and machine learning-based approaches. Among them, rule-based processing is the most common anomaly detection method, and the definition of rule is mainly divided into two categories: (1) algorithmic automatic extraction, which is a relatively simple extraction method and mainly detects anomalous data through pre-set extraction rules, such as EDE-FRMiner [18]; (2) expert knowledge extraction, which manually specifies rules by experts and then determines whether the data are anomalous or not. However, rule-based approaches usually faced: (1) limited by the experience of experts, there are large deviations in detection results due to insufficient specified rules; (2) the rule base needs to be updated in time, otherwise new anomalies cannot be detected in time; (3) the overhead of detection using matching rules is high. In the statistical-based approaches, the measured data need to obey a certain distribution and use the data for parameter estimation. The simpler methods include box plot, Crubbs test, 3σ criterion, etc., and the more complex ones include vector autoregressive (VAR), autoregressive moving level search (ARMA), etc.; However, the above methods are more suitable for low-dimensional data and need the measured data to meet assumed distribution. Machine learning-based approaches are composed of supervised learning [25], unsupervised learning [16] and weakly supervised learning [44] approaches. The difference between them lies in the labeling strength of the measured data. Although machine learning-based approaches can achieve good performance in many tasks [22], [6], [9], they are extremely challenging to apply due to the lack of high-quality labels in the measured dataset in most real cases and their inability to autonomously learn the features of data.

2.2. Deep Learning-based Anomaly Detection Approaches

Deep unsupervised learning-based approach [29] and deep weakly supervised learning-based approach [26], [29] had excellent ability to obtain complex internal relationships from unlabeled data or small portions with finitely labeled data, i.e., they address the challenging problem of high quality labeling in machine learning. For example, convolutional neural network (CNN) has been widely used in anomaly detection for its advantages of hierarchical feature extraction and translation invariance. On the basis of CNN, different modifications of CNN have been widely developed, such as neural heuristic analysis and sonar analysis using ROI (Region of Interest). Neural heuristic analysis [23] combines the concepts of neural networks and heuristics to solve complex optimization problems, it uses the neural network models to learn from data and develops the heuristic methods for decision-making or problem-solving tasks; In addition, it also adjusts and improves their performance based on experience and feedback through utilizing the learning ability of neural networks. Sonar analysis using ROI [34] is used to analyze the specific areas in sonar images or data that are expected to contain important information or features, it improves the efficiency and accuracy of sonar target detection or image segmentation via combining ROI analysis with cellular neural networks.

In particular, to be more relevant to the context of anomaly detection on time-series data, this subsection focuses on deep learning-based approaches on time-series data. Specifically, deep learning-based approaches are mainly composed of prediction-based and reconstruction-based [13] approaches. Prediction-based approach determines the presence of anomalies by predicting the data at the next timestamp, while reconstruction-based method encodes and decodes the normal data by deep neural networks to capture patterns in normal data as well as detects the anomalous data.

The current prediction-based approaches mainly include: anomaly detection approaches based on RNN or its variants (e.g., DeepLSTM [8]), CNN-based approaches (e.g., DeepAnt [29]), GNN-based approaches (e.g., GDN [15]), HTM-based approaches (e.g., RADM [31]), and Transformer-based approaches (e.g., SAND [37]). The above methods capture the timing information in the time-series data through different structures of the model, but generally use the forward description of time-series data to preprocess it to generate a sequence that fits the model. Such methods have the problem of ineffectiveness in dealing with some time-series data containing bidirectional semantic information.

Reconstruction-based approaches are mainly classified into: AE-based approaches (e.g., USAD [2]), VAE-based approaches (e.g., LSTM-VAE [33]), and GAN-based approaches (e.g., MAD-GAN [24]). These approaches first encode normal data based on the reconstruction principle and use deep neural networks to map it into a low-dimensional space, they then decode the encoding result into the original space to reconstruct an approximate original data as well as measure the abnormal degree of data by computing the differences between reconstructed data and original ones. In contrast, the self-encoders or variational self-encoders are often used as deep neural networks due to their compression and reconstruction capabilities. These methods do not require a priori knowledge or over-labeled data and can learn anomalous patterns directly from unlabeled data, but they have limitations such as low sensitivity to assumptions about normal data distribution and data imbalance. On other hand, the GAN-based anomaly detection methods use generators to generate the data similar to normal data as well as use discriminators to distinguish generated data from real data, thereby identifying the anomalous data by computing the differences between generated and real data. This category of anomaly detection approach has better generalization ability and stronger robustness, but has the problems such as unstable model training and susceptibility to noisy data.

In summary, the detection performance of most popular and advanced deep learning-based approaches (e.g., DAGMM [45], CAE_M [42], MTAD-GAT [43], MAD-GAN [24], GDN [15], USAD [2], DTAAD [41], TranAD [39] faces significantly deficient when dealing with the time-series data with bidirectional semantics due to the failure to consider the bidirectional semantic of the data. To better detect the anomalous data from time-series data, we propose an efficient anomaly detection approach called BiTCN-MHA based on BiTCN and MHA mechanism, which breaks the bottleneck of anomaly detection by introducing the bidirectional model, MHA mechanism and ELU activation function. The comparison of BiTCN-MHA and reviewed anomaly detection methods are shown in Table 1.

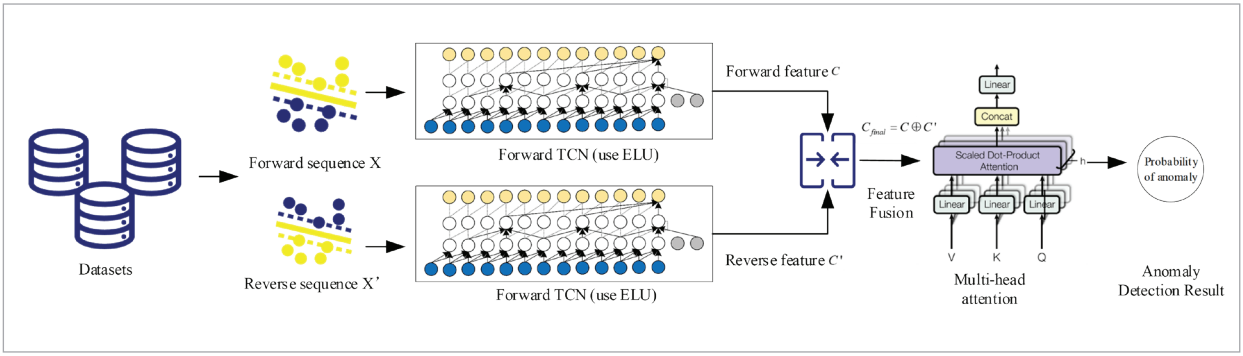
Table 1
The comparative characteristic of anomaly detection approaches

Models	Used activation function	Feature weight learning
DAGMM [45]	Tanh	×
CAE_M [42]	Sigmoid	×
MAD-GAN [24]	LeakyReLU	×
GDN [15]	LeakyreLU+Sigmoid	√
USAD [2]	ReLU+Sigmoid	×
DTAAD [41]	Sigmoid	√
MTAD-GAT [43]	LeakyReLU	×
TranAD [39]	Sigmoid	×
BiTCN-MHA (proposed)	ELU	√

3. Anomaly Detection Method BiTCN-MHA

In this section, we elaborate the proposed BiTCN-MHA approach based on BiTCN and MHA (Multi-Head Attention) mechanism, which consists of three main components: bi-bidirectional TCN model, ELU activation function and MHA mechanism.

Figure 1
The framework of BiTCN-MHA approach



3.1. The Framework of BiTCN-MHA

This subsection introduces the framework of BiTCN-MHA, and it is shown in Figure 1.

Firstly, the time-series data are processed into two input forms satisfying the forward TCN model and the reverse TCN model by preprocessing operations. Secondly, the processed input sequences are fed into the forward and reverse TCN models to extract the forward and reverse features of the data (containing all features of the data but not one feature), and the activation function of ELU is used to solve the neuron "death" problem during the learning process. Then, the forward features (C) and reverse features (C') of network traffic are fused using Equation (7) to provide more features for MHA layer. Next, the MHA mechanism is applied to give larger weights to the features that cause the data to be judged as anomalous, thereby avoiding the overload problem of feature information in the learning process and thus improving the detection efficiency. Finally, the probability of each data being anomaly data is output as the result. The details of each stage are described in Subsections 3.2-3.5.

3.2. Preprocessing of Time-series Data

The time-series data is a set of data objects with temporal sequence marked by the timestamp T, it is shown in Equation (1).

$$S = \{x_1^{(i)}, x_2^{(i)}, ..., x_T^{(i)}\} \tag{1}$$

where $x_1^{(i)}$ is the first value of i^{th} dimension of time-series data, T is the temporal length of time-series data with T^{th} timestamps. Meanwhile, for the anomaly detection process, the existing approach defines it as the prediction of subsequent h steps using sliding windows for the input training timing data T, i.e., the prediction of $\{x_{1+h}^{(i)}, x_{2+h}^{(i)}, ..., x_{T+h}^{(i)}\}$ is based on the $\{x_1^{(i)}, x_2^{(i)}, ..., x_T^{(i)}\}$, and then compare the difference between the true value and $\{x_{1+h}^{(i)}, x_{2+h}^{(i)}, ..., x_{T+h}^{(i)}\}$ with the pre-set threshold to determine whether the time-series data contains anomalies.

To further improve the detection efficiency, the same time-series data preprocessing as DTAAD is used, i.e., normalized training data and test data. In addition, a 50dB white noise was added to the data as data enhancement and sliced into the sliding time window, the specific preprocessing-related equations is shown in Equations (2)-(4).

$$\overline{x_t} \leftarrow \frac{x_t - min(\tau)}{max(\tau) - min(\tau) + \varepsilon} \tag{2}$$

$$SNR = 10\log_{10}\frac{\sum_1^N (\overline{x_t})^2}{\sum_1^N (\varepsilon_t)^2} \tag{3}$$

$$x_t = \overline{x_t} + \frac{\varepsilon_t}{100} \tag{4}$$

where x_t and ε_t denote the signal and noise, respectively, and the ratio of signal to noise is in dB. The $max(\tau)$ and $min(\tau)$ mean the maximum vector and minimum vector in the training time series τ. The ε is a very small constant to prevent overflow by division by zero for the entire input data in the range (0,1).

Through the above data pre-processing operation, the processed data is used as the training set for training BiTCN-MHA model.

3.3. BiTCN Model

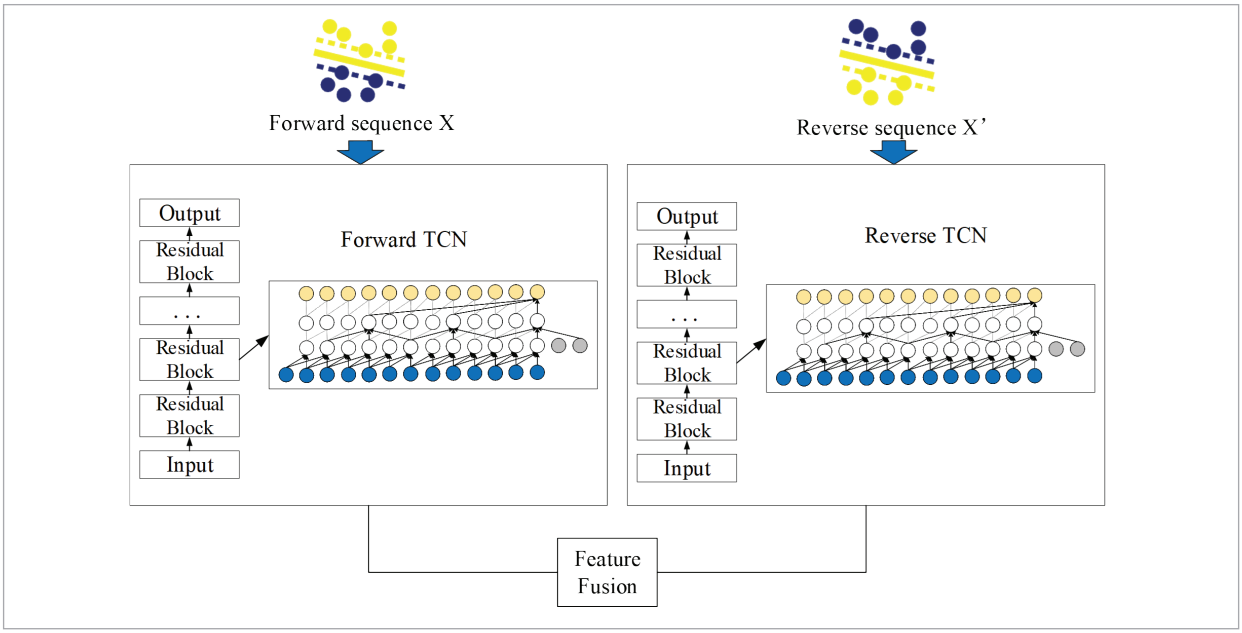
TCN model is usually used for processing the sequential data, it is composed of many stacking residual blocks. Each residual block consists of a causal convolution module, an inflation convolution module and a residual connection. Among them, the causal convolution module ensures the output of current moment only having the connection with the previous moments but not influenced by later moments by restricting the movement direction of convolution kernel, which helps TCN to obtain more accurate information as well as reduce the model parameters and computation when processing the time-series data.

Compared to the deep neural network models with similar functions such as RNN, LSTM and GRU, TCN is more efficient in processing time-series data for the following reasons. (1) TCN contains only convolutional and pooling layers, it has a simpler structure is easy to implement and accelerate the training speed; (2) TCN captures the long-term dependencies by setting up a "causal convolution module" to maintain the temporality of the time-series data during the convolution process; (3) TCN enhances the modeling ability by stacking multiple convolutional layers and uses techniques such as residual join and batch normalization to enhance the training effect. Numerous experiments were confirmed that TCN can achieve better results when dealing with the time-series data [41], [10], [11].

In order to further improve the detection efficiency on time-series data, a bidirectional TCN model (BiTCN, including forward TCN and inverse TCN) is used to learn the forward and inverse features of time-series data, respectively. Both forward and inverse TCN models use a one-dimensional fully convolutional neural network (1D-FCN) to ensure that each hidden layer has the same characteristics as the input layer and applies a generalized convolution operation to guarantee the output only correlated with the time step value in the previous layer and earlier time steps, i.e., the 1D-FCN and causal convolutional model are used to process the time-series data. The structure of proposed BiTCN model formed by the combination of forward and inverse TCNs is shown in Figure 2.

In addition, the interval sampling and the setting of different expansion coefficients are used to obtain larger sensory fields, and thus reducing the training time and improving the detection efficiency. Usually, the expansion factor of the model is set to $d=2n-1$, where n means the number of convolutional layers.

Figure 2
The structure of BiTCN model



With the increasing number of layers, the interval sampling distance becomes larger, a larger perceptual field can be obtained, i.e., capturing more temporal information. In particular, the subsequent dimension is set to 1 to describe the preprocessing of time-series data more conveniently, and the overall preprocessing of high-dimensional time-series data is the same as that of one-dimensional ones. Specifically, the one-dimensional sequence $X=(x_1, x_2,..., x_T)$ containing temporal information can be obtained after preprocessing, where T also represents the timestamp length, and the expansion convolution operation in TCN is shown in Equation (5).

$$F(X) = (X *_d f)(X) = \sum_{i=0}^{k-1} f(i) \cdot X_{s-d.i} \quad (5)$$

In Equation (5), d is the expansion coefficient, k is the size of convolution filter, and s-$d.i$ represents the information based on the index of d pointing to the past, i.e., the past direction. The forward feature extraction against the time-series data can be accomplished by Equation (2). Similarly, the one-dimensional sequence $X=(x_1, x_2,..., x_T)$ is inverted to generate the one-dimensional sequence $X'=(x_T, x_{T-1},..., x_1)$, and the generated X' is adopted to extract the reverse features through inflated convolution operation like forward sequence.

Finally, X and $F(X)$ are passed together into the next layer of network structure using residual connections to avoid information loss when passing between layers. Specifically, X and $F(X)$ are nonlinearly transformed with the use of ELU, which will be described in Subsection 3.4.

3.4. Activation Function of ELU

The commonly used activation function in TCN is an unsaturated one called ReLU. Although ReLU has the advantages of fast computation, effective in gradient disappearance, and no saturation for the positive part, it has some significant disadvantages: (1) The gradient is 0 when the input is negative, resulting in no activation (death) of neurons, which let the model cannot learn and update; (2) When the input value is too large, the output of activation function becomes very large, i.e., the neuron "explosion" problem occurs, which leads to network instability; (3) ReLU is a non-zero mean function, which may cause the output of some neurons to be always negative, thus reducing the model fitting ability.

In this paper, we adopt the ELU instead of ReLU to solve the neuron "death" problem of the ReLU, the structure of ELU is shown in Equation (6). When $x>0$, the ELU function is a linear function with a gradient of 1, while the ELU can take negative values when $x\leq 0$, where a is usually 1. The ELU function does not simply set all negative axis information to "0", but retains the negative axis gradient. Therefore, ELU effectively alleviates the gradient disappearance (i.e., neuron "death") problem. In addition, since ELU is a one-sided saturation function, it has a faster convergence speed compared to the activation functions such as ReLU, LeakyReLU, PReLU, etc.

$$ELU(X) = \begin{cases} x, & x > 0 \\ a(e^x - 1), & x \leq 0 \end{cases} \quad (6)$$

3.5. Multi-head Attention Mechanism

Although changing the unidirectional TCN model to a bidirectional structure and using the ELU can improve the anomaly detection to some extent. However, the model treats all features of time-series data as equally important during the training process, which allows unimportant features affecting the effectiveness of important features for anomaly detection. The attention mechanism, which prevents information overload and assigns different weights to different features, can be used in the training of models to effectively enhance the detection efficiency. Therefore, this paper applies the MHA mechanism to the BiTCN model.

Firstly, the extracted forward and reverse features are fused according to Equation (7), and the fused features are then added into the MHA mechanism, where C and C' represent the forward and reverse features, respectively, and C_{final} represents the fused features.

$$C_{final} = C \oplus C' \quad (7)$$

Then, the fused features C_{final} are linearly transformed according to the number of heads h to obtain h sets of three vectors Q, K, and V, i.e., query vector, key vector, and value vector. The Q, K and V are mapped to a low-latitude space according to Equation (8) to obtain Q', K' and V'. The W_Q, W_K and W_V are the linear transformation matrices of Q, K and V, respectively,

d_{model} denotes the dimension of original input vector, d_k denotes the dimension of linearly transformed vector, and Q', K' and V' are the mapped query vector, key vector and value vector, respectively.

$$\begin{cases} W_Q \in R^{d_{model} \times d_k}, & Q \in R^{n \times d_{model}}, & Q' = QW_Q \\ W_K \in R^{d_{model} \times d_k}, & Q \in R^{n \times d_{model}}, & K' = KW_K \\ W_V \in R^{d_{model} \times d_k}, & Q \in R^{n \times d_{model}}, & V' = VW_V \end{cases} \quad (8)$$

Finally, the attention of Q', K' and V' are calculated by Equation (9), and the h-group calculated results are concatenated in the last dimension to form the final results of MHA mechanism.

$$Attention(Q', K', V') = \text{softmax}(\frac{Q' K'^T}{\sqrt{d_k}})V' \quad (9)$$

3.6. Training Process of BiTCN-MHA

After introducing the details BiTCN model, ELU activation function and MHA mechanism, the training process of BiTCN-MHA based on bidirectional TCN and MHA mechanism is shown in Algorithm 1. Firstly, the pre-processed features X are reversed, and the features of X and X' are input into the BiTCN model. Then, pooling the extracted forward and reverse features through fully connected layer, and the obtained features are nonlinearly varied and spliced by the ELU activation function. Finally, the spliced features are processed for model training according to Equations (8)-(9).

Algorithm 1: BiTCN-MHA

Input: Time-series data $Data_{Time}$, Learning rate L, Batch b, Epoch e, Dropout D
Output: Detection result R

01. X=DataProcessing ($Data_{Time}$)
02. X'=*reverse*(X)
03. **for** each e **do**
04. **for** each b **do**
05. extract times-series features according to L and D
06. *lable*‘←classification result based on TCN
07. calculate loss function
08. renew weight values W_1 and W_2
09. get forward and reverse features c and c'
10. **end for**
11. **end for**
12. h=*ELU*(W_1*c+b_1), h'=*ELU*(W_2*c'+b_2)
13. H=(*softmax*(h)+*softmax*(h'))/2
14. R=multi-headattention(H)
15. **return** R

The computational complexity of proposed BiTCN-MHA algorithm is $O(h_1+h_2)$, it mainly depends on the computational complexity of data processing $O(h_1)$ and BiTCN operations $O(h_2)$. Among them, $O(h_1)$ depends on the dimension d of the preprocessed temporal data, $O(h_2)$ is generally represented as the product of three variables k, l and m^2, where k represents the size of a kernel, l represents the input length, m represents the number of output channels or feature maps in the convolutional layer.

4. Experiments and Analysis

The BiTCN-MHA model is compared with eight state-of-the-art anomaly detection models (including DAGMM [45], CAE_M [42], MTAD-GAT [43], MAD-GAN [24], GDN [15], USAD [2], DTAAD [41], TranAD [39]) on six datasets in this section to verify its detection effectiveness, and then the ablation experiments are conducted to demonstrate the necessity of each part of BiTCN-MHA model.

The running environment of the experiment is Win 10 with two I7-10700 2.90GHz CPUs, one NVIDIA GeForce RTX 3060 Ti GPU and the BiTCN-MHA is realized in python 3.10.6.

4.1. The Introduction of Time-series Datasets

In the experiment, we use six datasets (including SWaT [27], SMAP [21], MBA [28], UCR [14], NAB [1], SMD [38]) to verify the efficiency of BiTCN-MHA method, and the details of the used datasets are shown in Table 2. In the experiment, the ratio of training datasets and testing datasets is chosen as 4:1.

1. SWaT (Secure Water Treatment): The data is generated by a continuously operating water treatment system, each row of data contains a time stamp and the corresponding sensor/actuator measurements.
2. SMAP (Soil Moisture Active Passive): The data is collected from Soil Moisture Remote Sensing developed by NASA, and the anomalies are extracted from anomaly reports generated by spacecraft detection systems.
3. MBA (MIT-BIH Supraventricular Arrhythmia): This dataset is the first standard metric for arrhythmia detector evaluation, it has been used over 500 basic tests.
4. UCR (HexagonML): This dataset is composed of multivariate time series data obtained from real-world sources.

Table 2
The information about the used datasets

Datasets	Entities (Dimensions)	Numbers	Anomalies(%)
SwaT	51 (1)	946719	11.98
SMAP	25 (55)	562800	13.13
MBA	2 (8)	200000	0.14
UCR	1 (4)	7500	1.88
NAB	1 (6)	8066	1.92
SMD	38 (4)	1416840	4.16

5 NAB (Numenta Anomaly Benchmark): This dataset has 50 different time-series, each dataset contains a csv file and a label file containing anomalous time periods in the time series.

6 SMD (Server Machine Dataset): This dataset is collected from a large Internet company within five weeks, it records relevant information generated by the server.

4.2. Evaluation Metrics

Aimed at evaluating the efficiency of BiTCN-MHA approach, we used four evaluation metrics (including *P* (Precision), *R* (Recall), AUC value and F_1-*score*) in the experiment, their definitions are shown in Equations (10)-(13).

$$AUC = \frac{\sum_{i=1}^{m-1}(x_{i+1} - x_i) * y_i}{(m-1) * (1 - y_0)} \tag{10}$$

$$P = \frac{TP}{TP + FP} \tag{11}$$

$$R = \frac{TP}{TP + FN} \tag{12}$$

$$F_1 \text{-} score = \frac{2 * P * R}{P + R} \tag{13}$$

where m is the total number of positive and negative samples, x_i and y_i are the horizontal and vertical coordinates of the i^{th} point on ROC curve, and y_i is the intercept of ROC curve on the x-axis. The model has better model performance when AUC value is much closer to 1, while the model has worse performance when the AUC value is close to 0.5. In Equations (11)-(13), true positive (*TP*) represents the number of correctly detected samples, false negative (*FN*) represents the number of samples from a category that are incorrectly detected to other categories, false positive (*FP*) represents the number of samples that are incorrectly detected.

4.3. Detection Efficiency of BiTCN-MHA model

The BiTCN-MHA model is compared with eight baselines on six publicly available time-series datasets in this section, we repeat the experiment for 50 times and calculate the average experimental result, it is shown in Table 3.

As is shown in Table 3 that the proposed BiTCN-MHA model can obtain better precision, recall, F_1-score and AUC metrics compared with these eight models on most datasets. On the datasets UCR and SMD, the AUC metric of DAGMM model achieves slightly higher AUC metric than BiTCN-MHA model, it is owing to that the DAGMM using deep autoencoding Gaussian mixture model to learn the features of datasets, which can effectively detect the potential anomalies in the datasets. Compared with other efficient approaches, the BiTCN-MHA model performs better on all six publicly available datasets, it is attributes the BiTCN-MHA model uses bidirectional TCN to learn the bidirectional features of the datasets as well as uses ELU to avoid the neuron "death" and MHA mechanism to provide different weights to different features based on their importance to the determining of anomalies, these three techniques used in the BiTCN-MHA can greatly improve the detection efficiency of anomalies. Specific, the CAE_M model performs worse in the nine compared models in most cases, while the F_1-score and AUC of other models are not stable on different datasets. Specially, on the NAB dataset, the proposed BiTCN-MHA model can achieve an average AUC of 0.93453, while the average AUC of DAGMM, CAE_M, MAD-GAN, GDN, USAD , DTADD, MTAD-GAT and TranAD are 0.75462 (23.84%↑), 0.66629 (40.26%↑), 0.81463 (14.72%↑), 0.79627 (17.36%↑), 0.66637 (40.24%↑), 0.86508 (8.03%↑), 0.90115 (3.7%↑), 0.81225 (15.05%↑), the proposed BiTCN-MHA model has a largest increase on the AUC metric; Similarly, the proposed BiTCN-MHA model can achieve an average F_1-Score of 0.89206, while the average F_1-Score of DAGMM, CAE_M, MAD-GAN,

Table 3
Detection efficiency of BiTCN-MHA and state-of-the-arts on six time-series datasets

Methods	Metrics	SWaT	SMAP	MBA	UCR	NAB	SMD
DAGMM	P	0.97895	0.87853	0.93471	0.57843	0.77412	0.90017
	R	0.69459	0.75849	0.93954	0.59293	0.51015	0.89771
	F_1-score	0.81261	0.81411	0.93712	0.58559	0.61501	0.89894
	AUC	0.84250	0.95328	0.95158	**0.99852**	0.75462	**0.98855**
CAE_M	P	0.96991	0.82340	0.83745	0.71233	0.78771	0.91167
	R	0.63048	0.82476	0.96706	0.77534	0.62744	**0.91920**
	F_1-score	0.76420	0.82408	0.89760	0.74250	0.69850	0.91542
	AUC	0.83851	0.95202	0.94570	0.98772	0.66629	0.93085
MAD-GAN	P	0.94775	0.82338	0.92174	0.87741	0.86514	**0.99970**
	R	**0.70744**	0.75366	0.88674	0.75062	0.60677	0.85444
	F_1-score	0.81015	0.78698	0.90390	0.80908	0.71328	0.92138
	AUC	0.84050	0.94569	0.94879	0.99459	0.81463	0.86565
GDN	P	0.97412	0.77842	0.85662	0.69183	0.81301	0.71174
	R	0.69047	0.91343	**0.99736**	0.92224	0.61325	0.64241
	F_1-score	0.80813	0.84054	0.92165	0.79059	0.69914	0.67530
	AUC	0.83904	0.95572	0.96251	0.99660	0.79627	0.94796
USAD	P	**0.99886**	0.74730	0.88799	0.89412	0.84211	0.98541
	R	0.67774	0.92298	0.97642	0.80767	0.54741	0.86398
	F_1-score	0.80755	0.82590	0.93011	0.84870	0.66351	0.92071
	AUC	0.83861	0.94731	0.94675	0.99082	0.66637	0.97542
DTAAD	P	0.94770	0.84599	0.96637	0.88884	0.88796	0.87884
	R	0.68083	0.91199	0.99540	0.92527	0.82349	0.89409
	F_1-score	0.79240	0.87775	0.98067	0.90669	0.85451	0.88640
	AUC	0.83702	0.96916	0.97392	0.98924	0.86508	0.98536
TranAD	P	0.97458	0.82770	0.95697	**0.91076**	**0.88888**	0.99743
	R	0.69381	0.97244	0.95759	0.98129	0.85623	0.90653
	F_1-score	0.81057	0.89425	0.95728	0.94471	0.87225	0.91987
	AUC	0.84227	0.96541	0.96747	0.97354	0.90115	0.98258
MTAD-GAT	P	0.97190	0.79770	0.91192	0.78034	0.84276	0.81198
	R	0.69102	**0.99988**	0.93562	0.95191	0.69689	0.90921
	F_1-score	0.80774	0.88742	0.92362	0.85763	0.76291	0.85785
	AUC	0.83554	0.95112	0.95431	0.99651	0.81225	0.90510
BiTCN-MHA	P	0.98527	**0.89514**	**0.99881**	0.90243	0.88412	0.92447
	R	0.69318	0.99508	0.97752	**0.99999**	**0.90014**	0.91903
	F_1-score	**0.81381**	**0.94247**	**0.98805**	**0.94871**	**0.89206**	**0.92174**
	AUC	**0.84406**	**0.97160**	**0.98572**	0.99784	**0.93453**	0.98390

GDN, USAD, DTADD, MTAD-GAT and TranAD are 0.61501 (45.05%↑), 0.69850 (27.71%↑), 0.71328 (25.06%↑), 0.69914 (27.59%↑), 0.66351 (34.45%↑), 0.85451 (4.39%↑), 0.87225 (2.27%↑), 0.76291 (16.93%↑). The improvement of BiTCN-MHA on other datasets is similar with that on dataset NAB. In general, extensive experimental results verify that the BiTCN-MHA model can accurately detect anomalies from time-series data.

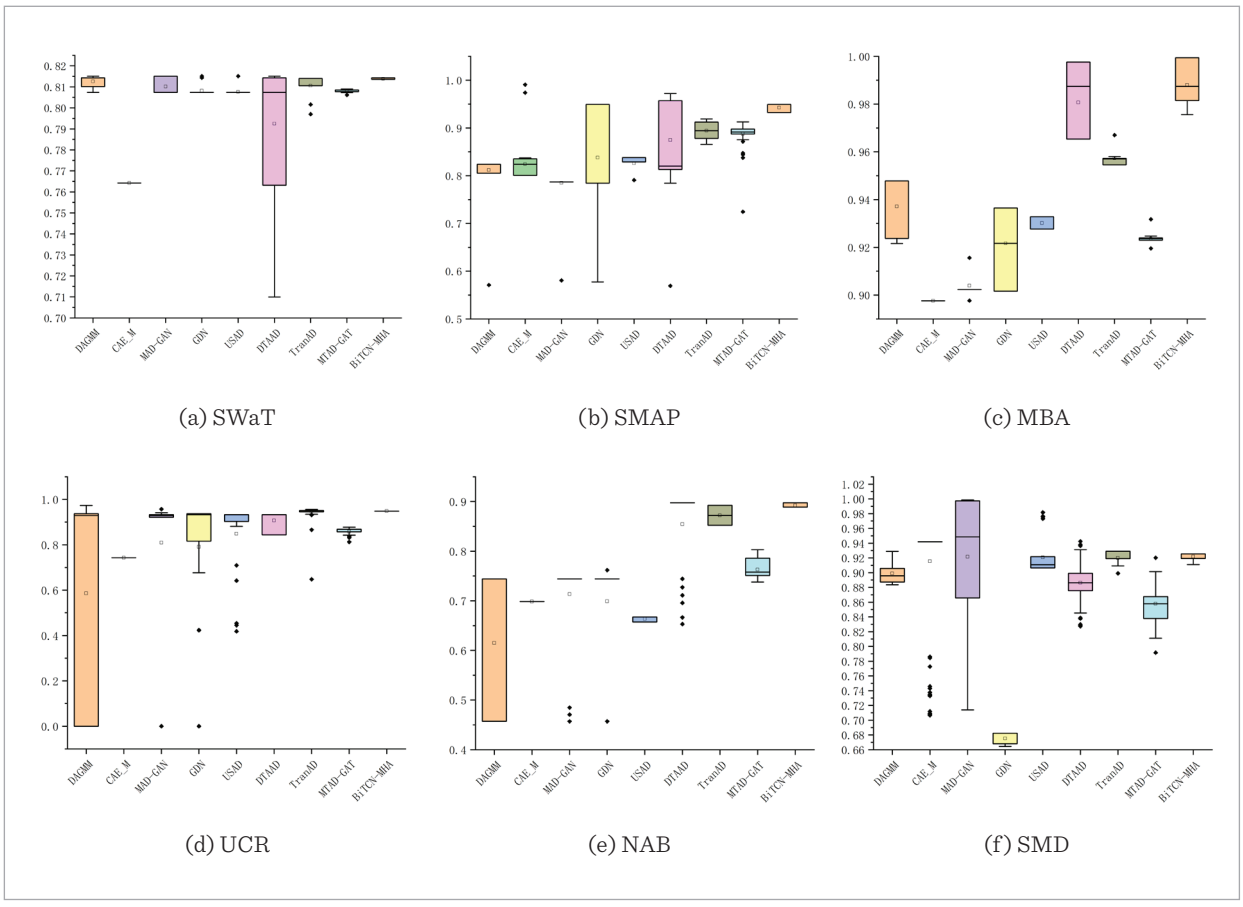
4.4. Stability of BiTCN-MHA Model

In addition to the detection efficiency, the stability is an important factor to measure the detection performance, therefore, we perform experiments to verify whether the BiTCN-MHA model has stability compared with other models on metrics of F_1-score and AUC. The experiment is also run for 50 times, and the final result is presented in Figures 3-4.

It can be seen from Figures 3-4 that the BiTCN-MHA model has no outlier in both F_1-score and AUC metrics. Except for the AUC metric of BiTCN-MHA on the MBA dataset that is slightly lower than DTAAD in some experiments, and the F_1-score metric of BiTCN-MHA on the SMD dataset that is slightly lower than MAD-GAN in some experiments, the F_1-score metric as well as the AUC metric of BiTCN-MHA model are higher than compared anomaly detection models.

As presented in Figures 3(a)-(f) and Figures 4(a)-(f), the proposed BiTCN-MHA model has a relatively short interquartile range, which means that the BiTCN-MHA model has a stable detection performance. Although the interquartile range of detection results of the BiTCN-MHA model on some time-series datasets is slightly larger than the range of compared methods, but the detection efficiency is higher, and the BiTCN-MHA model does not show any outli-

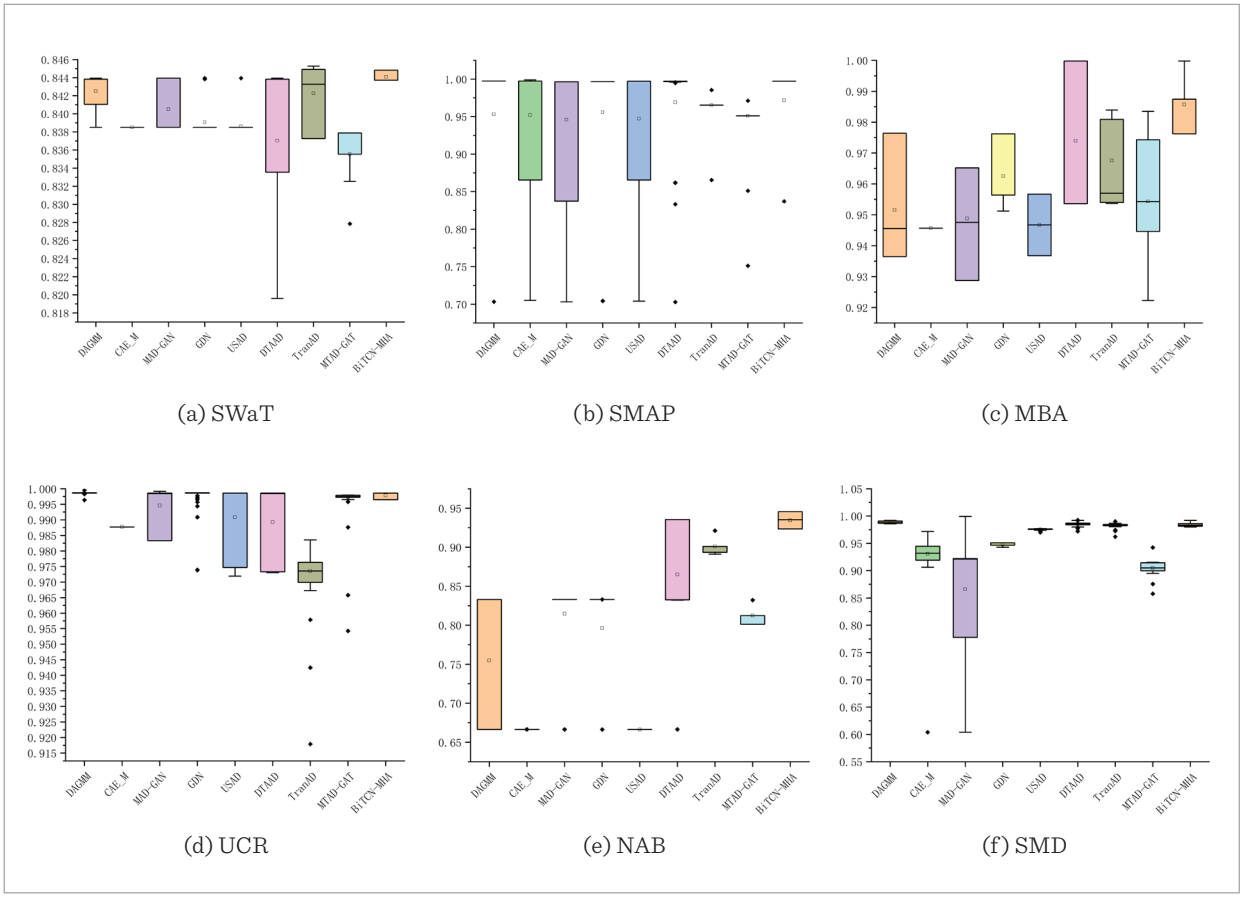
Figure 3
The stability of F1-score metric on compared anomaly detection methods



(a) SWaT (b) SMAP (c) MBA

(d) UCR (e) NAB (f) SMD

Figure 4
The stability of AUC metric on seven anomaly detection methods



ers, which also indicates BiTCN-MHA model having higher detection performance and better stability.

In comparison with eight state-of-the-arts, the BiTCN-MHA model uses the bidirectional TCN model to forward and backward learn the time-series data, thereby better grasping the bidirectional features of time-series data; In addition, we change the habit of using ReLU in the TCN model and apply the ELU to avoid neuron "death", therefore, the BiTCN-MHA model does not converge quickly and thus achieving better convergence. In the BiTCN-MHA model, we also incorporate the MHA mechanism to set higher weights to more important features, thus allowing the detection model to focus on those features that are more closely associated with the anomalies. These improvements can better enable the BiTCN-MHA model to perform well in anomaly detection.

4.5. Ablation Experiments

Influence of bidirectional TCN structure: We investigate the performance of BiTCN-MHA with the unidirectional TCN-MHA model on six datasets, and the experimental result is shown in Table 4. Compared with the unidirectional structure, the use of BiTCN in the detection of anomalies can improve the detection efficiency. The reason is that the learning of bidirectional semantic features can promote BiTCN to identify and capture the correlation of features, which can enable the detection model to understand the relationship between features more accurately and facilitate the extraction of richer and more informative feature representations, thus improving the accuracy and robustness of the model. This also clearly demonstrates the advantages of including bidirectional feature learning in the proposed BiTCN-MHA model.

Table 4
Detection efficiency of BiTCN-MHA method without some components

Methods	Metrics	SWaT	SMAP	MBA	UCR	NAB	SMD
With unibidirectional structure	*P*	**0.96541**	0.95772	0.95124	0.94545	0.78874	0.87667
	R	0.66318	0.89583	0.97374	**0.91751**	**0.99960**	**0.93817**
	F_1-score	0.78625	0.92574	0.96236	0.93127	0.88174	0.90638
	AUC	0.81238	0.93893	0.96451	0.95374	0.90392	0.94562
With ReLU	*P*	0.85228	**0.99981**	0.99475	0.99112	0.78412	**0.99986**
	R	0.71015	0.84636	0.91489	0.87373	0.99296	0.82131
	F_1-score	0.77475	0.91671	0.95315	0.92873	0.87627	0.90183
	AUC	0.79548	0.93074	0.96748	0.94127	0.89528	0.93562
Without MHA	*P*	0.78942	0.98713	0.91948	0.97359	0.79657	0.97885
	R	0.70687	0.84237	0.96647	0.87875	0.94711	0.82661
	F_1-score	0.74587	0.90902	0.94239	0.92374	0.86534	0.89631
	AUC	0.77513	0.92407	0.95317	0.93672	0.88725	0.92582
BiTCN-MHA	*P*	0.87221	0.96987	**0.99666**	**0.99901**	**0.94276**	0.99583
	R	**0.76274**	**0.91658**	**0.97959**	0.90323	0.84653	0.85791
	F_1-score	**0.81381**	**0.94247**	**0.98805**	**0.94871**	**0.89206**	**0.92174**
	AUC	**0.84406**	**0.97160**	**0.98572**	**0.99784**	**0.93453**	**0.98390**

Influence of ELU activation function: We investigate the effect of BiTCN-MHA model under the ReLU activation function as well as the ELU activation function. Table 4 shows the detection results, where "With ReLU" represents the results under using ReLU. Table 4 shows that the detection efficiency with the use of ELU is higher than that with the use of ReLU activation function in both F_1-score and AUC metrics. This is due to the fact that the ELU activation function effectively mitigates the gradient disappearance problem and has a faster convergence rate due to the fact that ELU is a one-sided saturation function. Compared with ELU, ReLU activation function has poor fitting ability, easy to fall into gradient "explosion" and neuron "death" as well as unsaturated for negative numbers.

Influence of MHA mechanism: We also compare the effectiveness of using the MHA and without using the MHA model for anomaly detection on six publicly available datasets, and the final result is presented in Table 4. It is shown in Table 4 that the BiTCN-MHA model outperforms the model without MHA in both F_1-score and AUC metrics; therefore, the use of MHA mechanism in the anomaly detection model can improve the detection efficiency, which is caused by that the models incorporating MHA mechanism simultaneously focus on different features in the input sequence, which can learn a richer and more comprehensive feature representation; In addition, each attention head focus on different semantic information, which allows the model to better understand the structure and semantics associated with the input sequence, thereby being more robust to input perturbations.

In general, extensive ablation experiments show that the bidirectional structure of TCN model, the ELU activation function and the MHA mechanism are essential for the BiTCN-MHA model. The simultaneous use of these three components in BiTCN-MHA can effectively improve the detection accuracy and stability for the time-series data.

5. Conclusion

In this paper, we propose an anomaly detection approach called BiTCN-MHA based on BiTCN and MHA mechanism. BiTCN-MHA mainly relies on the

BiTCN model to extract the temporal features in the time-series data, it first normalizes and adds noise to the original data to improve its quality, and then feds the data after pre-processing into the forward TCN model and reverse TCN model to extract the forward and reverse features of data more effectively. And then, the ELU activation function is used instead of ReLU to solve the problem of neuron "death". In addition, it also uses the MHA mechanism to set larger weight to important features to avoid the overload problem of feature information, thereby enhancing the detection efficiency of abnormal data. To evaluate the effectiveness of BiTCN-MHA approach, we conduct massive experiments on six publicly-available datasets. Extensive results present that the BiTCN-MHA improves precision, recall, AUC and F1-Score by about 6.10%, 10.16%, 4.06% and 8.50% respectively on six datasets. The stability experimental result also shows that BiTCN-MHA approach can accurately detect the abnormal data with high stability. In addition, the ablation experimental results verify that each component of BiTCN-MHA can improve the detection efficiency of abnormal data, and the combination of BiTCN model, ELU activation function and MHA mechanism are the best choose in the detection of abnormal data.

Although the proposed BiTCN-MHA method outperforms the eight state-of-the-arts on several publicly available datasets and performs more consistently on most of the datasets, its stability is not as good as the compared methods on some publicly available datasets. Therefore, we would like to change the loss function as well as introduce some depth modules (e.g., residual network models) in the future to further upgrade the detection efficiency and stability of the BiTCN-MHA method.

Acknowledgements

This work was partly supported by the Shanxi Agricultural University Youth Science and Technology Innovation Fund (Grant no. 2019017).

References

1. Ahmad, S., Lavin, A., Purdy, S., Agha, Z. Unsupervised Real-time Anomaly Detection for Streaming Data. Neurocomputing, 2017, 262, 134-147. https://doi.org/10.1016/j.neucom.2017.04.070
2. Audibert, J., Michiardi, P., Guyard, F., Marti, S., Zuluaga, M.A. USAD: Unsupervised Anomaly Detection on Multivariate Time Series. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, 3395-3404. https://doi.org/10.1145/3394486.3403392
3. Cai, S. H., Chen, J. F., Chen, H. B., Zhang, C., Li, Q., Sosu, R. N. A., Yin, S. An Efficient Anomaly Detection Method for Uncertain Data Based on Minimal Rare Patterns with the Consideration of Anti-monotonic Constraints. Information Sciences, 2021, 580(1), 620-642. https://doi.org/10.1016/j.ins.2021.08.097
4. Cai, S. H., Huang, R. B., Chen, J. F., Zhang, C., Liu, B., Yin, S., Geng, Y. An Efficient Outlier Detection Method for Data Streams Based on Closed Frequent Patterns by Considering Anti-monotonic Constraints. Information Sciences, 2021, 555(1), 125-146. https://doi.org/10.1016/j.ins.2020.12.050
5. Cai, S. H., Li, S. C., Yuan, G., Hao, S. B., Sun, R. Z. Mi-FI-Outlier: Minimal Infrequent Itemset-based Outlier Detection Approach on Uncertain Data Stream. Knowledge-Based Systems, 2020, 191(5), 105268-105289. https://doi.org/10.1016/j.knosys.2019.105268
6. Canizo, M., Triguero, I., Conde, A., Onieva, E. Multi-head CNN-RNN for Multi-time Series Anomaly Detection: An Industrial Case Study. Neurocomputing, 2019, 363, 246-260. https://doi.org/10.1016/j.neucom.2019.07.034
7. Cao, N., Lin, Y. R., Gotz, D., Du, F. Z-Glyph: Visualizing Outliers in Multivariate Data. Information Visualization, 2018, 17(1), 22-40. https://doi.org/10.1177/1473871616686635
8. Chauhan, S., Vig, L. Anomaly Detection in ECG Time Signals via Deep Long Short-Term Memory Networks. In 2015 IEEE International Conference on Data Science and Advanced Analytics, 2015, 1-7. https://doi.org/10.1109/DSAA.2015.7344872
9. Chen, A., Fu, Y., Zheng, X., Lu, G. An Efficient Network Behavior Anomaly Detection Using a Hybrid DBN-LSTM Network. Computers & Security, 2022, 114, 102600. https://doi.org/10.1016/j.cose.2021.102600
10. Chen, J. N., Chong, W. T., Yu, S. Y., Xu, Z. Tan, C. H., Chen, N. J. TCN-based Lightweight Log Anomaly Detection in Cloud-edge Collaborative Environment. In

10th International Conference on Advanced Cloud and Big Data, 2022, 13-18. https://doi.org/10.1109/CBD58033.2022.00012

11. Cheng, Y. L., Xu, Y., Zhong, H., Liu, Y. HS-TCN: A Semi-supervised Hierarchical Stacking Temporal Convolutional Network for Anomaly Detection in IoT. In 2019 IEEE 38th International Performance Computing and Communications Conference, 2019, 1-7. https://doi.org/10.1109/IPCCC47392.2019.8958755
12. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. Empirical Methods in Natural language Processing, 2014, 1724-1734. https://doi.org/10.3115/v1/D14-1179
13. Darban, Z. Z., Webb, G. I., Pan, S., Aggarwal, C. C., Salehi, M. Deep Learning for Time Series Anomaly Detection: A Survey. CoRR, 2022. https://doi.org/10.48550/arXiv.2211.05244
14. Dau, H. A., Bagnall, A., Kamgar, K., Yeh, C. C. M., Zhu, Y., Gharghabi, S., Ratanamahatana, C. A., Keogh, E. The UCR Time Series Archive. IEEE/CAA Journal of Automatica Sinica, 2019, 6(6), 1293-1305. https://doi.org/10.1109/JAS.2019.1911747
15. Deng, A., and Hooi, B. Graph Neural Network-Based Anomaly Detection in Multivariate Time Series. In Proceedings of the AAAI Conference on Artificial Intelligence, 2021, 35(1), 4027-4035. https://doi.org/10.1609/aaai.v35i5.16523
16. Ergen, T., Kozat, S. S. Unsupervised Anomaly Detection with LSTM Neural Networks. IEEE Transactions on Neural Networks and Learning Systems, 2020, 31(8), 3127-3141. https://doi.org/10.1109/TNNLS.2019.2935975
17. Goldstein, M., Score, A. D. H. B. O. A Fast Unsupervised Anomaly Detection Algorithm. KI-2012: Poster and Demo Track, 2012, 1(1), 59-63.
18. Guendouzi, W., Boukra, A. A New Differential Evolution Algorithm for Cooperative Fuzzy Rule Mining: Application to Anomaly Detection. Evolutionary Intelligence, 2022, 15(4), 2667-2678. https://doi.org/10.1007/s12065-021-00637-3
19. Gupta, M., Gao, J., Aggarwal, C. C., Han, J. Outlier Detection for Temporal Data: A Survey. IEEE Transactions on Knowledge and Data Engineering, 2013, 26(9), 2250-2267. https://doi.org/10.1109/TKDE.2013.184
20. Hochreiter, S., Schmidhuber, J. Long Short-term Memory. Neural Computation, 1997, 9(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735
21. Hundman, K., Constantinou, V., Laporte, C., Colwell, I., Soderstrom, T. Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, 387-395. https://doi.org/10.1145/3219819.3219845
22. Kong, F., Li, J., Jiang, B., Wang, H. H., Song, H. B. Integrated Generative Model for Industrial Anomaly Detection via Bidirectional LSTM and Attention Mechanism. IEEE Transactions on Industrial Informatics, 2023, 19(1), 541-550. https://doi.org/10.1109/TII.2021.3078192
23. Lakshmi, R. K.,, Aravapalli, R. S. Novel Heuristic-Based Hybrid ResNeXt with Recurrent Neural Network to Handle Multi-class Classification of Sentiment Analysis. Machine Learning-Science and Technology, 2023, 2632-2153. https://doi.org/10.1088/2632-2153/acc0d5
24. Li, D., Chen, D. C., Jin, B. H., Shi, L., Goh, J., Ng, S. K. MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks. In International Conference on Artificial Neural Networks. Springer, 2019, 11730, 703-716. https://doi.org/10.1007/978-3-030-30490-4_56
25. Liu, J., Song, X., Zhou, Y., Peng, X., Zhang, Y., Liu, P., Wu, D., Zhu, C. Deep Anomaly Detection in Packet Payload. Neurocomputing, 2022, 485, 205-218. https://doi.org/10.1016/j.neucom.2021.01.146
26. Liu, Y., Liu, J., Zhao, M., Li, S., Song, L. Collaborative Normality Learning Framework for Weakly Supervised Video Anomaly Detection. IEEE Transactions on Circuits and Systems II: Express Briefs, 2022, 69(5), 2508-2512. https://doi.org/10.1109/TCSII.2022.3161061
27. Mathur, A. P., Tippenhauer, N. O. SWaT: A Water Treatment Testbed for Research and Training on ICS Security. In 2016 International Workshop on Cyber-physical Systems for Smart Water Networks, 2016, 31-36. https://doi.org/10.1109/CySWater.2016.7469060
28. Moody, G. B., Mark, R. G. The Impact of the MIT-BIH Arrhythmia Database. IEEE Medicine and Biology Magazine, 2001, 20(3), 45-50. https://doi.org/10.1109/51.932724
29. Munir, M., Siddiqui, S. A., Dengel, A., Ahmed, S. DeepAnT: A Deep Learning Approach for Unsupervised Anomaly Detection in Time Series. IEEE Access, 2018, 7(1), 1991-2005. https://doi.org/10.1109/ACCESS.2018.2886457
30. Munoz, A., Muruzábal, J. Self-Organizing Maps for Outlier Detection. Neurocomputing, 1998, 18(1-3), 33-60. https://doi.org/10.1016/S0925-2312(97)00068-4

31. Nan, D., Gao, H. B., Bu, H. Y., Ma, H. X., Si, H. W. Multivariate-Time-Series-Driven Real-time Anomaly Detection Based on Bayesian Network. Sensors, 2018, 18(10), 3367-3379. https://doi.org/10.3390/s18103367
32. Pang, G., Shen, C., Cao, L., Hengel, A. V. D. Deep Learning for Anomaly Detection: A Review. ACM Computing Surveys, 2021, 54(2), 1-38. https://doi.org/10.1145/3439950
33. Park, D., Hoshi, Y., Kemp, C.C. A Multimodal Anomaly Detector for Robot-Assisted Feeding Using an LSTM-Based Variational Autoencoder. IEEE Robotics and Automation Letters, 2018, 3(3), 1544-1551. https://doi.org/10.1109/LRA.2018.2801475
34. Polap, D., Wawrzyniak, N., Sielicka, M. W. Side-Scan Sonar Analysis Using ROI Analysis and Deep Neural Networks. IEEE Transactions on Geoscience and Remote Sensing, 2022, 1-8. https://doi.org/10.1109/TGRS.2022.3147367
35. Roberts, S., Osborne, M., Ebden, M., Reece, S., Gibson, N., Aigrain, S. Gaussian Processes for Time-Series Modelling. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 2013, 371(1984), 20110550-20110575. https://doi.org/10.1098/rsta.2011.0550
36. Rumelhart, D. E., Hinton, G. E., Williams, R. J. Learning Representations by Back-Propagating Errors. Nature, 1986, 323, 533-536. https://doi.org/10.1038/323533a0
37. Song, H., Rajan, D., Thiagarajan, J. J., Spanias, A. Attend and Diagnose: Clinical Time Series Analysis Using Attention Models. In Proceedings of the AAAI Conference on Artificial Intelligence, 2018, 32(1), 4091-4098. https://doi.org/10.1609/aaai.v32i1.11635
38. Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., Pei, D. 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, 2019, 2828-2837. https://doi.org/10.1145/3292500.3330672
39. Tuli, S., Casale, G., Jennings, N. R. TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data. Proceedings of VLDB, 2022, 15(6), 1201-1214. https://doi.org/10.14778/3514061.3514067
40. Wu, Y., Hernández-Lobato, J. M., Zoubin, G. Dynamic Covariance Models for Multivariate Financial Time Series. In Proceedings of the 30th International Conference on International Conference on Machine Learning, 2013, 28(3), 558-566. https://dl.acm.org/doi/10.5555/3042817.3042999
41. Yu, L. DTAAD: Dual TCN-Attention Networks for Anomaly Detection in Multivariate Time Series Data. CoRR, 2023. https://doi.org/10.2139/ssrn.4410420
42. Zhang, Y., Chen, Y., Wang, J., Pan, Z. Unsupervised Deep Anomaly Detection for Multi-Sensor Time-Series Signals. IEEE Transactions on Knowledge and Data Engineering, 2021, 35(2), 2118-2132. https://doi.org/10.1109/TKDE.2021.3102110
43. Zhao, H., Wang, Y., Duan, J., Huang, C., Cao, D., Tong, Y., Xu, B., Bai, J., Tong, J., Zhang, Q. Multivariate Time-Series Anomaly Detection via Graph Attention Network. In 2020 IEEE International Conference on Data Mining, 2020, 841-850. https://doi.org/10.1109/ICDM50108.2020.00093
44. Zhou, Y., Song, X., Zhang, Y., Liu, F., Zhu, C., Liu, L. Feature Encoding with Autoencoders for Weakly Supervised Anomaly Detection. IEEE Transactions on Neural Networks and Learning Systems, 2022, 33(6), 2454-2465. https://doi.org/10.1109/TNNLS.2021.3086137
45. Zong, B., Song, Q., Min, M. R., Cheng, W., Lumezanu, C., Cho, D., Chen, H. Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection. In 6th International Conference on Learning Representations, 2018, 1-19.