Hindawi
Complexity
Volume 2020, Article ID 8846608, 9 pages
https://doi.org/10.1155/2020/8846608

Research Article

MTAD-TF: Multivariate Time Series Anomaly Detection Using the Combination of Temporal Pattern and Feature Pattern

Q. He,[1] **Y. J. Zheng**,[1] **C.L. Zhang**,[1,2] **and H. Y. Wang**[1]

[1]*School of Science, Beijing University of Civil Engineering and Architecture, Beijing 100044, China*
[2]*Beijing Advanced Innovation Center for Future Urban Design, Beijing University of Civil Engineering and Architecture, Beijing 100044, China*

Correspondence should be addressed to Q. He; heqiang@bucea.edu.cn

Received 26 August 2020; Revised 23 September 2020; Accepted 16 October 2020; Published 29 October 2020

Academic Editor: Tongqian Zhang

Currently, multivariate time series anomaly detection has made great progress in many fields and occupied an important position. The common limitation of many related studies is that there is only temporal pattern without capturing the relationship between variables and the loss of information leads to false warnings. Our article proposes an unsupervised multivariate time series anomaly detection. In the prediction part, multiscale convolution and graph attention network are mainly used to capture information in temporal pattern with feature pattern. The threshold selection part uses the root mean square error between the predicted value and the actual value to perform extreme value analysis to obtain the threshold. Finally, the model in this paper outperforms other latest models on actual datasets.

1. Introduction

Anomaly detection of time series data has always been a hot issue in academia and industry. The detection of abnormal points and the location of abnormal areas can provide important information at critical moments, so that people can intervene with abnormal events in a targeted way to prevent or eliminate abnormal events. Anomaly detection of time series data has attracted people's attention in industry, finance, military, medical treatment, insurance, robotics, multiagent, network security, IOT, complex biological systems, etc. [1, 2].

The anomaly detection of time series is to detect points with outliers, oscillations, or other abnormal conditions. In general, the proportion of anomalies in the overall time series is very low, so people hope to successfully capture the outliers by learning the distribution of original data or other characteristics through the algorithm. Univariate anomaly detection is carried out on the time series with only one feature. Since there is only one dimension of data, many traditional filtering algorithms can be used, that is, spectral residual algorithm [3]. Multivariate time series anomaly detection refers to the anomaly detection of time series data with multiple sequences. This kind of problem is extended based on univariate time series anomaly detection. The occurrence of anomalies in multivariate time series data is often determined by multiple features, and the individual analysis of each feature cannot accurately locate the anomalies. Complex biological systems generally have this characteristic. For example, time series data from an epidemic model may include the number of patients, the number of healthy people, infection rate and the immunization rate, etc. The severity of epidemic cannot be judged by partial characteristics. Therefore, a more reasonable method is to comprehensively analyse multiple variables to identify anomalies.

At present, significant progress has been made in the study of MTAD (multivariate time series anomaly detection) in deep learning. For example, Malhotra et al. [4] proposed an encoder-decoder network based on LSTM, which modelled the reconstruction probability of "normal" time series and used reconstruction errors to detect anomalies in multiple sensors. Hundman et al. [5] used the long- and short-time memory network (LSTM) to detect the spacecraft

multivariate time series based on prediction loss. Ding et al. [6] proposed RADM, a real-time anomaly detection algorithm based on Hierarchical Temporal Memory (HTM) and Bayesian Network (BN), which improved the performance of real-time anomaly detection. However, most of the proposed methods often rely on the RNN (Recurrent Neural Network) learning properties and distribution in temporal pattern; relationship between sequences is still unutilized. Therefore, we believe that new latent dependencies can be exploited from feature pattern, which is more conductive to anomaly detection. We propose a method combination of temporal pattern and feature pattern.

Our main contribution is as follows:

(1) To the best of our knowledge, this is the first study on multivariate time series anomaly detection generally from a graph-based perspective with graph attention network in forecast

(2) We propose a new model that combines temporal with feature pattern, capturing more latent relationship between variables

(3) Experimental results show that our method outperforms the state-of-the-art methods on 3 benchmarks

The arrangement of this article is as follows. We give related work on time series anomaly detection in Section 2. In Section 3, the prerequisite knowledge of GAT and GRU in the model is introduced. In Section 4, the proposed method is introduced in detail. The fifth section conducts experiments and analysis. Finally, we summarize the full text.

2. Related Work

Anomaly detection is also known as novelty detection, outlier detection, or event detection in other related fields [7]. Time series anomaly detection is one of the most concerning problems. It can be classified into supervised, semisupervised, and unsupervised abnormal detection according to whether labels are used during training. Supervised learning method [8] requires labelled data for training and can only identify known abnormal types [9], so its application scope is limited. Semisupervised method is a kind of learning method combining supervised learning and unsupervised learning. Semisupervised method uses a large amount of untagged data as well as tagged data, rarely studied in the field of TSAD (Time Series Anomaly Detection). Therefore, research of TSAD focuses on the unsupervised problem.

According to the number of sequences in the data, the problem can be divided into univariate and multivariate time series anomaly detection. Univariate time series anomaly detection [3, 10, 11] only considers whether the variables conforms to long-term pattern; when there is a big difference between data value and the overall distribution, it is regarded as an outlier instance. The traditional method in univariate time series anomaly detection is to use mainly hand-made features to model patterns of normal and abnormal events [12]. For example, there are SVD [13], wavelet analysis [9], ARIMA [14], and so on. Besides, Netflix released a document based on robust Principal Component Analysis [15] and received a good response. Twitter also published a method which uses the seasonal hybrid extreme study deviation test (S-H-ESD) [16]. In addition, the use of neural networks for detection has also made great progress [17]. Multivariable problems have multiple variables on each timestamp [18]. The existing multivariate time series anomaly detection methods can be divided into two categories: (1) univariate based anomaly detection [15], where each sequence is monitored separately by univariate algorithm and the results are summarized to give the final judgment, and (2) direct anomaly detection [19], where multiple features are considered at the same time for algorithm analysis. Let us focus on the second type of approach. Zong et al. [20] proposed a model which uses deep autoencoder to generate low dimensional data, represent the reconstruction errors of each input data point, and input into a Gaussian mixture model (GMM) for multivariable anomaly detection. LSTM-VAE algorithm [7] is a LSTM network based on encoder-decoder to reconstruct the error of time series and use the reconstruction error to detect the abnormal situation of some sensors. LSTM-NDT [5] is an unsupervised algorithm without parameter threshold selection. The objective of this paper is to establish an anomaly detection system to monitor the data sent back by the spacecraft which is marked by experts in related fields.

Graph neural network is very popular in recent years which have enjoyed great progress in dealing with spatial dependencies among entities in a network. Gugulothu et al. [21] combined nontime pattern reduction technology and periodic automatic encoder through the end-to-end learning framework for time series modelling. OmniAnomaly [22] proposes a stochastic recurrent neural network that captures the normal pattern of multiple variable through modelling data distribution with stochastic variables.

3. Preliminaries

3.1. Problem Statement. When analysing real-world datasets, a common requirement is to find out those instances that can be considered as outliers, which are significantly different from most other points. The goal of the anomaly detection task is to be data-driven to find abnormal of all samples. In our work, we are concerned about multivariable data $X = \{x_1, x_2, \ldots, x_N\} \in \mathbb{R}^{m*n}$; the value at time i is $x_i \in \mathbb{R}^m$, $i = 1, 2, \ldots, n$. m means there are m variables and n is the length of data. Our target is to determine whether x_t is an abnormal point. This is a time series problem; we have a huge amount of data; historical data is helpful for understanding the current moment x_t. To efficiently use and learn the information of X, sliding window w: $x_{t-w}, x_{t-w+1}, \ldots, x_{t-1}$ used to predict x_t which would be considered to be normal. The difference between the predicted x_t with the ground truth will be put into the threshold selection module; the larger the difference, the greater the possibility of x_t being abnormal; when such difference exceeds the threshold we set, we consider it to be an abnormality.

3.2. Basics of GAT and GRU

3.2.1. GAT (Graph Attention Network). We know that many data are in Euclidean space. The most significant characteristic of data in Euclidean space is that it has a regular spatial structure. For example, the picture is a regular square grid, the voice data is a one-dimensional sequence, and so on. These data can be represented by a one-dimensional or two-dimensional matrix. However, many data in real life do not have a regular spatial structure, that is, data in non-Euclidean space, such as abstract graphs of electronic transactions, recommendation systems, social networks, and so on; each node in the graph is related to other nodes. The connection is not fixed. Therefore, people use graph neural networks to model data in non-Euclidean spaces. In recent years, due to the strong expressiveness of graph structure, the research of analysing graphs with machine learning methods has received more and more attention. Graph neural network (GNN) is a method of processing graph pattern information based on deep learning. Due to its better performance and interpretability, GNN has become a widely used graph analysis method. Commonly used graph neural networks include Graph convolution networks, graph attention networks, and graph autoencoder. Among them, GAT [23] proposes to utilize the attention mechanism to add weighted features of neighbouring nodes. The weight of neighbouring node features completely depends on the node, independent of the graph structure. In our model, to find the latent relationship between variables, we use GAT to calculate the correlation between nodes. The specific details are explained in Section 4.3.

3.2.2. GRU (Gated Recurrent Unit). Recurrent neural network (RNN) is a kind of neural network that captures the dynamic information in serialized data through the periodic connection of nodes in the hidden layer. It is different from feedforward neural networks; RNN can save the state of a context and even store, learn and express relevant information in any long context window. No longer limited to the spatial boundaries of traditional neural networks, it can be extended in time series. Intuitively speaking, there is an edge between the nodes of the hidden layer of this time and the hidden layer of the next moment. But RNN's most significant drawback is that it cannot learn to preserve and exploit older information, namely, gradient vanishing and gradient explosion. Sepp Hochreiter and Jurgen Schmidhuber proposed long- and short-term memory (LSTM) in 1997 [24]. LSTM is a kind of periodic neural network, which alleviates the problem of RNN to some extent. Practice shows that this method is very suitable for processing time series data. In fact, the LSTM algorithm has evolved many variations in recent years. Rafal Jozefowicz et al. of Google conducted a comprehensive architecture search to evaluate over 10,000 different RNN/LSTM architectures [25] and as a result we could not find an architecture with better performance than the GRU, and, except for the language model, GRU works better than LSTM in other application scenarios. GRU (Gated Recurrent Unit) is a variant of LSTM, which has fewer parameters and is more efficient than LSTM. Hence, our model chooses GRU structure instead of LSTM.

Cho et al. [26] proposed a Gated Recurrent Unit (GRU) to enable each recursive unit to adaptively capture the dependencies of different time scales. Like classical recurrent neural networks, GRU are a chain of neural units too. Its structure is expressed mathematically as follows:

$$\begin{aligned} r_t &= \sigma\left(W_r \cdot \left[h_{t-p}, x_t\right]\right), \\ z_t &= \sigma\left(W_z \cdot \left[h_{t-p}, x_t\right]\right), \\ \widetilde{h_t} &= \tanh\left(W_{\widetilde{h}} \cdot \left[r_t * h_{t-p}, x_t\right]\right), \; h_t = (1 - z_t) * h_{t-p} + z_t * \widetilde{h_t}, \; y_{ts} = \sigma\left(W_o \cdot h_t\right). \end{aligned} \tag{1}$$

x_t and h_{t-1} represent the input at the current time and the output $h_t = \tanh(W_{\widetilde{h}} \cdot [r_t * h_{t-p}, x_t])$ at the next time. Where r_t is a set of reset gates, it is used to control how much information about previous state is forgotten. The smaller the value of reset gate, the more the past information is discarded. z_t is update gates. The update gate is used to control the degree how much information from the previous moment is brought into the current state. The larger the value is, the more the information from the current needs to remain and the less the information from the previous neuron can be retained. (,) represents two vectors concatenate, and * is an element-wise multiplication.

σ is the commonly used sigmoid function which controls numbers between 0 and 1. We are accustomed to using tanh function (hyperbolic tangent function) as hidden update activation function:

$$\begin{aligned} \text{sigmoid: } y &= \frac{1}{\left(1 + e^{-x}\right)}, \\ \tanh: y &= \frac{\left(e^{x} - e^{-x}\right)}{\left(e^{x} + e^{-x}\right)}. \end{aligned} \tag{2}$$

4. Proposed Model

4.1. Model Architecture. As demonstrated in Figure 1, our framework consists of three core components: the temporal convolution model, the graph attention model, and threshold select model. The result obtained in the first two models is the forecasting of our MTAD-TF. The root mean square error (RMSE) between the forecasted result and the real value is input to the error threshold selection model. If the error exceeds the threshold which we set through POT, it is considered that an anomaly occurs at the moment.

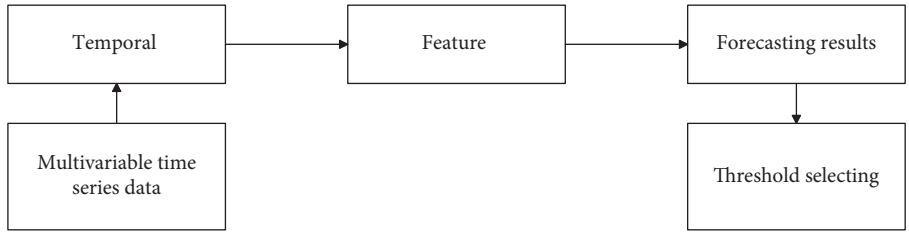

Figure 1: Overview of MTAD-TF.

The explanation of the forecasting model is as follows:

(i) Temporal convolution component: we propose a temporal convolution model to capture temporal patterns by multiscale 1D convolutions, which can find temporal patterns with multiple periods

(ii) Graph attention component: the graph attention network is used in the feature dimension; the interrelation between variables is beneficial to forecast the time series

4.2. Data Preprocessing. For a multivariable time series, the dimensions of different variables are quite different. We cannot allow these differences to affect subsequent prediction and threshold selection. Therefore, we preprocess the data with the maximum-minimum normalization method in both training subsets with testing subsets:

$$\widehat{x_t} = \frac{x_t - \min X_{\text{training}}}{\max X_{\text{training}} - \min X_{\text{training}}}. \tag{3}$$

4.3. Forecasting Model. The overview of the proposed model is shown in Figure 2. First, for the sake of alleviating the possible noise effects of the original data X, 1D convolution operation is carried out to smooth the data:

$$\begin{aligned} X_{\text{CNN}} &= \text{RELU}\left(W_{\text{CNN}} * X + b_{\text{CNN}}\right), \\ \text{RELU} &= \max\left(0, x\right). \end{aligned} \tag{4}$$

The result of convolution X_{CNN} is then fed into three identical blocks which are shown as green box. Each block has temporal convolution component in series with graph attention networks.

4.3.1. Temporal Convolution Component. The temporal convolution module captures sequential patterns of time series data in temporal dimension through 1D convolutional filters to come up with a temporal convolution module that is able to both discover temporal patterns with various ranges and handle long sequences, that is, using multiscale convolution filters [27]. However, how to choose the correct filter size is a challenging problem. To understand convolution in terms of communication theory and image processing, the convolution kernel size is generally set to odd [28]. The reasons are as follows: compared with even numbers, odd numbers have a center point and are more sensitive to edges and lines, which can extract edge information more effectively and avoid the deviation of position information. In addition, the odd number can ensure that the two sides of the image are symmetrical to each other when padding, so that size of the output image is the same as size of the input. Therefore, as shown in Figure 3, we select filters sizes of 1×3, 1×5, 1×7, and 1×9 which consist of temporal inception layer. The combination of these filters of different sizes can contain some periodic temporal signals, such as data of period 12. The model can start the input layer from the first temporal convolution layer through the 1×5 and then from the second temporal convolution layer through the 1×7. The selection of small convolution kernel can not only reduce the parameters but also add more nonlinear mappings to improve the robustness. Finally, we patch the results of different convolution, respectively, to restore the previous data size. The input of temporal convolution component in block 2 is the average value of GAT's output and X_{CNN}. TC component in block 3 is the average value of block 2's input (include X_{CNN}) and block 2's output.

4.3.2. Graph Attention Network Component. Multivariate time series anomaly detection is a challenge due to the increase of variable and data volume. However, more variable also means more information which is brought. It is actually very critical for anomaly detection. Previous models did not pay attention to feature pattern, but only focus on temporal pattern. Therefore, we combine temporal pattern and feature pattern in the model. Specially, each block has a temporal convolution component that connects to a GAT. In GAT, each node in the graph can be assigned different weights based on the characteristics of its neighbor nodes. And it does not require costly matrix operations or rely on a preconceived graph structure.

The input to the graph attention layer is a set of vectors for a node: $\{v_1, v_2, \ldots, v_n\}$, where v_i have the same dimension with x_i. The output of each node calculated by the GAT layer is shown as follows:

$$h_i = \sigma\left(\sum_{j=1}^{L} \alpha_{ij} v_j\right), \tag{5}$$

$$e_{ij} = \text{Leaky RELU}\left(W \cdot \left(v_i \oplus v_j\right)\right)^{\top}. \tag{6}$$

$$\text{Leaky RELU: } y_i = \begin{cases} x_i, & \text{if } x_i \geq 0, \\ \dfrac{x_i}{a_i}, & \text{if } x_i \leq 0, \end{cases} \quad a_i \in (1, \infty), \tag{7}$$

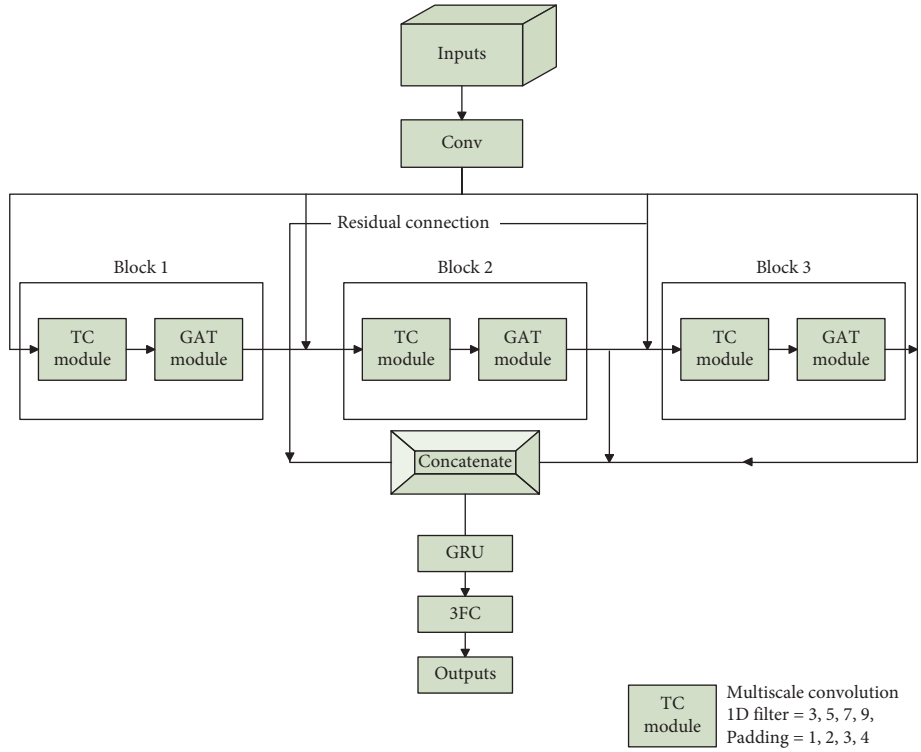

Figure 2: Forecasting model.

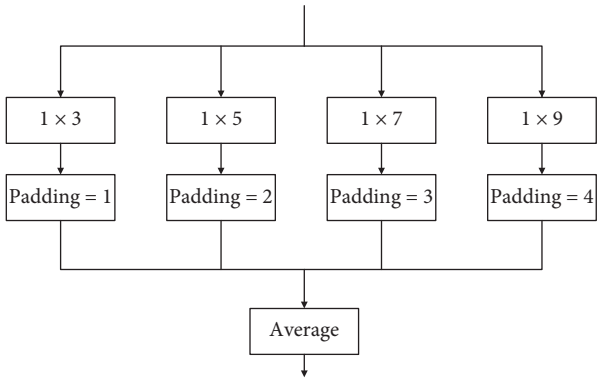

Figure 3: Temporal convolution component.

$$\alpha_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_{l=1}^{L} \exp\left(e_{ik}\right)}, \tag{8}$$

where h_i is the output of node x_i with the same dimension. α_{ij} is the correlation degree between x_i and x_j like (8) is calculated: $\oplus$ is the result of concatenate of two nodes, and w is the parameters obtained by learning. Leaky RELU is a nonlinear activation function as shown in (7). L denotes the number of adjacent points to x_i.

The results of each GAT and X_{CNN} (after 1D convolution of original input X) are the data of the same dimension, which are three-dimensional tensor, and each dimension is batch size, window size, and the number of variables, respectively. The output of GAT which is in three blocks and X are concatenated in the third dimension of tensor, which thickens the temporal information of data and is conducive to prediction from GRU. Finally, the results of the forecasting part are obtained by carrying on the three full connection layers.

4.4. Threshold Selection Model. The loss function of the prediction model selects root mean square error (RMSE) is as follows:

$$\mathrm{Loss}_{\mathrm{forecasting}}(t) = \sqrt{\sum_{i=1}^{m}\left(\widehat{y}_{t,i} - x_{t,i}\right)^2}, \tag{9}$$

where $\widehat{y}_{t,i}$ is the prediction value of the i-th feature at time t and $x_{t,i}$ is the real value at the same time. The RMSE between them denotes loss at time t.

The test set was input to the trained forecasting model, and the RMS loss between the predicted value and the true value of each observation point in the test set was recorded as $\left\{l_1, l_2, \ldots, l_Q\right\} \in \mathbb{R}^Q$ and utilizes POT (peaks over threshold) model of EVT (extreme value theory) to select the threshold value of the subsequence.

Extreme value theory is a statistical theory to find the law of extreme values in a sequence. It is generally believed that extreme values are the outliers to be found in the problem of anomaly detection, and they are located at the tail of the distribution in most cases. The advantage of the extreme value theory is that it does not need to assume the data distribution and the threshold can be set automatically through parameter selection. The second theorem POT shows that samples larger than threshold are subject to

generalized Pareto distribution (GPD). Therefore, select the threshold th through POT:

$$\overline{F}_{\text{th}} = P(L - \text{th} > l | L > \text{th}) \sim \left(1 + \frac{\gamma l}{\beta}\right)^{1/\gamma}, \tag{10}$$

where th is the initial threshold. γ denotes shape parameters in GPD and β is any value in scale parameters $L = \{l_1, l_2, \ldots, l_Q\}$. L-th represents the part above the threshold. th is the quantile obtained by experience. Similar to literature [10], we utilize maximum likelihood estimation (MLE) for parameter estimation of $\widehat{\gamma}$ and $\widehat{\beta}$. The threshold th_F is calculated according to the following formula:

$$\text{th}_F \simeq \text{th} - \frac{\widehat{\beta}}{\widehat{\gamma}}\left(\left(\frac{qQ}{Q_{\text{th}}}\right)^{-\widehat{\gamma}} - 1\right). \tag{11}$$

q is the proportion of $L > \text{th}$ and Q is the number of observed values. Q_{th} denotes the number of $L > \text{th}$. To select the threshold value of POT, the process of parameter adjustment is needed.

5. Experiment and Analysis

5.1. Benchmarks and Evaluation Metrics. Regarding datasets, we use three real-world datasets to verify the effectiveness of MTAD-TF, namely, MSL (Mars Science Laboratory) rover, SMAP (Soil Moisture Active Passive) satellite, and SMD.

MSL and SMAP are two public datasets of NASA's spacecraft [29].

SMD [22] is five weeks of server data in a large Internet company, which has been published on GitHub. SMD is divided into two parts with the same data size. The first part is the training set and the second part is the testing set. The abnormal data on the testing set has been marked by experts in related fields. Among them, the training set and the testing set contain 28 groups, which need to be trained and tested separately. That is, the model trained on the first group of data in the training set is tested by the same group of the testing set. The final score is the average of 28 groups.

The details of the three datasets are given in Table 1, including the number of variables, size of the training set and testing set, proportion of abnormal samples in the testing set, and partial variable names.

Regarding metrics, we followed the typical evaluation metrics like other anomaly detection models: precision, recall, and F1 score. They are defined as follows:

$$\begin{aligned} \text{precision}(P) &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ \text{recall}(R) &= \frac{\text{TP}}{(\text{TP} + \text{FN})}, \\ \text{F1} &= \frac{2 \times \text{precision} \times \text{recall}}{(\text{precision} + \text{recall})}. \end{aligned} \tag{12}$$

Among them, TP is true positives (correctly detected anomaly), FP represents false positives (falsely detected anomaly), and FN refers to false negative (falsely detected normally). The higher the values of the above three indicators, the stronger the robustness of the model.

5.2. Baselines for Comparison. This section will show the comparison results with the other 4 baselines on 3 benchmarks. The compared models include LSTM-NDT [5], LSTM-VAE [7], DAGMM [20], and OmniAnomaly [22]:

(i) LSTM-NDT: LSTM is used for anomaly detection of multidimensional time series which also is a dynamic and unsupervised method for determining threshold. Besides, to reduce the false positive rate and identify false positive data, a "pruning strategy" is proposed.

(ii) LSTM-VAE: VAE's feedforward network uses LSTM replacement but does not consider the dependence between stochastic variables.

(iii) DAGMM: combine neural network, estimation network, and Gaussian mixture model organically to do unsupervised anomaly detection.

(iv) OmniAnomaly: the core idea of this paper is to learn latent representations to capture the normal patterns of multivariate time series while considering time dependence and stochastic.

Table 2 summarizes the evaluation results of all the baselines, which shows excellent generalization capability and achieves the best F1 score on 4 datasets.

LSTM-NDT has a high score on SMAP, but it performs poorly on MSL and SMD, reflecting that the model is very sensitive to different scenarios. Our model is stable and has excellent performance on different benchmarks.

Short-term information is also very important for multivariable time series. The reason why DAGMM's performance is not ideal is that short-term information is not considered. We utilize multiscale convolution, which can better adapt to data with different periods. This article also conducts additional ablation experiments (see Section 5.3) to compare the effectiveness of different components in our model.

OmniAnomaly applies a stochastic model, regards variables as stochastic variables, and then learns its distribution, which has high performance on the three datasets. The limitation of this model is that it does not consider the relationship between the variables.

5.3. Ablation Study. To illustrate the necessity and effectiveness of core components in the forecasting part, we conduct an ablation study on the four datasets to validate the multiscale convolution, GAT, and GRU that contribute to the improved outcomes of our proposed model. Firstly, we name the MTAD-TF without different components as follows:

Table 1: Dataset information.

Dataset	MSL	SMAP	SMD
No. of attributes	55	25	38
Training subset size	58317	135183	708405
Testing subset size	73729	427617	708420
Anomaly rate (%)	10.72	13.13	4.16
Variables information	Telemetry data: computational, radiation, temperature, power, activities, etc.		CPU load, network usage, memory usage, etc.

Table 2: Performance of our model and baselines.

Dataset	MSL			SMAP			SMD		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
LSTM-NDT	0.5944	0.5374	0.5640	0.8965	0.8846	0.8905	0.5684	0.6438	0.6037
LSTM-VAE	0.5257	0.9546	0.6780	0.8551	0.6366	0.7298	0.7922	0.7075	0.7842
DAGMM	0.5412	0.9934	0.7007	0.5845	0.9058	0.7105	0.5835	0.9042	0.7093
OmniAnomaly	0.8867	0.9117	0.8989	0.7416	0.9776	0.8434	0.8334	0.9449	0.8857
MTAD-TF	0.9043	0.8988	0.9015	0.9779	0.8192	0.8916	0.9045	0.9048	0.8940

(i) w/o temporal: removing the multiscale convolution processing in the temporal pattern, only GAT is left in each block

(ii) w/o GAT: Removing the GAT processing in feature pattern, only temporal pattern is left in each block

(iii) w/o GRU: Removing GRU means X_{CNN} and output of three blocks are directly ingested to the FC layer

Table 3: Ablation study. F1 scores are reported.

	MSL	SMAP	SMD
MTAD-TF	0.9015	0.8916	0.8940
w/o temporal	0.7238	0.6945	0.7520
w/o GAT	0.6827	0.7089	0.6373
w/o GRU	0.8174	0.7000	0.7502

From Table 3, different components have different effect on different benchmarks. For MSL and SMD, deletion of GAT makes the F1 score drop the most, while SMAP is most affected by temporal convolution component. The score of EEG-EYE has not decreased much, but it has reduced to varying degrees.

5.4. Case Study. We will carry out case analysis of noise experiment in the EEG-EYE state data and GAT in this part.

EEG- (electroencephalogram-) EYE state is from UCI, one continuous EEG measurement with the Emotive EEG Neuroheadset, looking for the relationship between 13 EEGs in different positions of the human brain with the opening and closing of human eyes. Therefore, EEG-EYE state is a dataset that can be classified into two categories. We regard the open-eye label as the anomaly to be searched for and then perform anomaly detection on it.

5.4.1. Noise Experiment. To understand the antinoise ability of the model, we carried out case analysis of noise adding experiment. Five kinds of Gaussian white noise with mean value of 0 and variance of {0.1, 0.2, 0.3, 0.4, 0.5} were added into the training set, respectively. Then the trained model was tested with the unchanged test set, and the F1 value was obtained as shown in the blue broken line in Figure 4. As the variance of Gaussian noise increases, the data shows a downward trend, which conforms to our common sense. However, it also indicates that the model is still not robust enough and the addition of noise does not play a role in data enhancement. The effect of variance 0.02 is better than that of variance 0.01. Compared with variance 0.01, the noise of variance 0.02 increases the difficulty of network training, prevents overfitting, and improves the generalization ability, which can be regarded as the effect of data enhancement.

According to the verification in literature [10], it can be known that one-dimensional convolution has the effect of smoothing data. From another perspective, we illustrate the function of 1D convolution with experimental scores, and we add a contrast experiment to the above pure noise experiment: noise with different variances is added to the model with 1D convolution removed. As shown in the orange broken line in Figure 4, compared with the score in pure noise, the score of without convolution drops significantly, indicating that the existence of convolution can reduce the impact of noise during data preprocessing.

5.4.2. GAT. We took out the correlation between abnormal and normal before the abnormality from GAT, respectively, and drew the heat map in Figure 5. The right side of Figure 5 shows the correlation between feature 1 and features 2, 3, 4, 5, 6, 16, 17, 18, 19, 20, and 21 at normal time, while the correlation was at abnormal time on the left. The darker the color block, the higher the correlation between features, and vice versa. On the same horizontal line, the large chromatic aberration between the left and right sides means that when an abnormality occurs, the correlation between features has

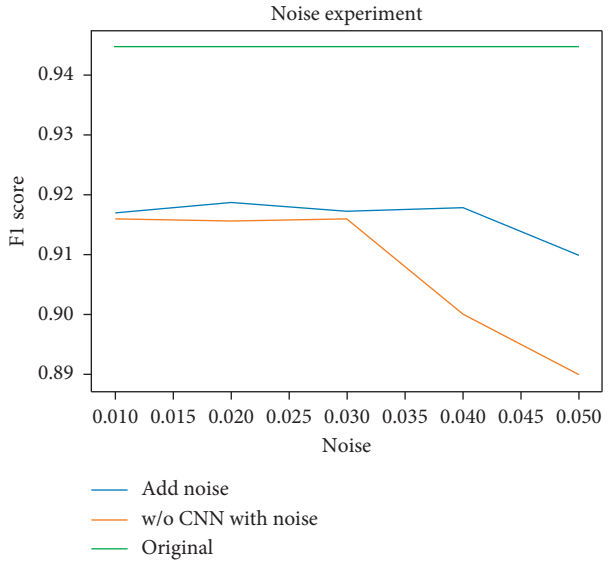

Figure 4: Noise experiment. The green line is the score of the original model without any processing; the value is as high as 0.945. The blue line is the score of the noise with different variances. The orange line is the score of the model without 1D convolution in data preprocess as well as added noises.

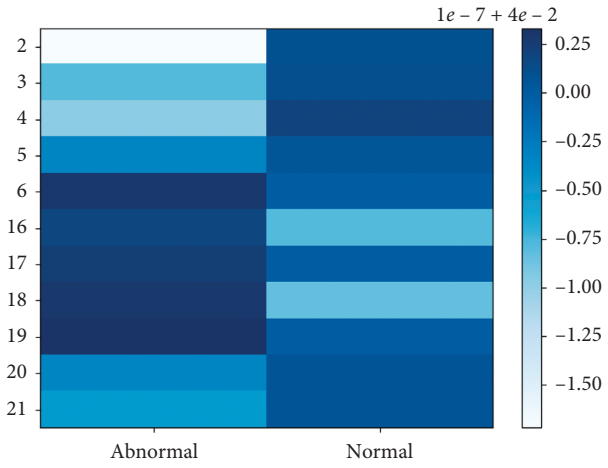

Figure 5: Heat map about correlation between variables. The number of the chromaticity bar is the chromaticity value, not the correlation between the features.

changed greatly, which can be used as a partial basis for abnormal location. Due to the lack of information about abnormal location in the dataset, further experimental verification cannot be carried out. However, it can be assumed that when an abnormality occurs, the correlation between certain features is significantly different from normal conditions.

6. Conclusions

In this paper, a new multivariate time series anomaly detection framework MTAD-TF is proposed. By using the temporal pattern and feature pattern model of multiple time series to make joint prediction, more latent information can be obtained than that of single pattern model. The method is superior to the other four baselines in the three common datasets. In addition, this model has a good antinoise ability and the GAT maybe can help with abnormal location. Future work may come from two aspects. First, attempts to combine the prediction model with the reconstruction model may further improve the accuracy of the model. Secondly, there is too little information on abnormal location and it is hoped that further abnormal location experiments can be carried out to improve the robustness of the MTAD-TF.

Appendix

A.1. Notation X

A is batch of multivariate time series input. x is an instance of X m number of variables (feature) in every instance. w is the length of X in sliding window. $\hat{x}_i$ is an instance output after data preprocessing. v_i is input node representation for a GAT layer. v_i is input node representation for a GAT layer. h_i is output node representation for a GAT layer. α_{ij} is attention score of node j to node i in a GAT layer. k_0 is the size of filter in 1D convolution. k_1 is hidden dimension of the GRU layer in forecasting component. k_2 is hidden dimension of 3 fully connected layers in forecasting component.

A.2. Experimental Settings

We use the same sliding window $w = 100$ in SMAP and SMD. w is for MSL and EEG-EYE state is set to 120 and 50, respectively. The size of filter in 1D convolution we use in all datasets is $k_0 = 7$. $k_1 = k_2 = 150$ in all dataset except EEG-EYE state which is 100. We use the Adam optimizer to train our model for 100 epochs with an initial learning rate 0.001.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (nos. 62072024 and 61473111), Projects of Beijing Advanced Innovation Center for Future Urban Design, Beijing University of Civil Engineering and Architecture (nos. UDC2019033324 and UDC2017033322), Scientific Research Foundation of Beijing University of Civil Engineering and Architecture (no. KYJJ2017017), Natural Science Foundation of Guangdong Province (no. 2018A0303130026), and Natural Science Foundation of Hebei Province (no. F2018201096).

References

[1] S. Ahmad and S. Purdy, "Real-time anomaly detection for streaming analytics," 2016, http://arXiv.org/abs/1607.02480.

[2] S. Reza and H. Javad, "Automatic support vector data description," *Soft Computing*, vol. 22, no. 1, pp. 147–158, 2018.

[3] B. Harvald, E. Schaumburg, Y. J. Wang et al., "Time-series anomaly detection service at microsoft," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, vol. 138, no. 7, pp. 420–422, Anchorage, AK, USA, August-2019.

[4] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," *European Symposium on Artificial Neural Networks*, vol. 89, 2015.

[5] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstorm, "Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 387–395, London, UK, August 2018.

[6] N. Ding, H. Gao, H. Bu, H. Ma, and H. Si, "Multivariate-time-series-driven real-time anomaly detection based on bayesian network," *Sensors*, vol. 18, no. 10, p. 3367, 2018.

[7] P. Daehyung, H. Yuuna, and K. Charles, "A multimodal anomaly detect or for robot-assisted feeding using an LSTM-based variational autoencoder," 2017, http://arXiv.org/abs/1711.00614.

[8] A. Rodriguez, D. Bourne, M. Mason et al., "Failure detection in assembly: force signature analysis," in *Proceedings of the IEEE Conference on Automation Science & Engineering*, pp. 210–215, September 2010.

[9] W. Lu and A. A. Ghorbani, "Network anomaly detection based on wavelet analysis," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, pp. 1–16, 2009.

[10] T.-Y. Kim and S.-B. Cho, "Web traffic anomaly detection using C-LSTM neural networks," *Expert Systems With Applications*, vol. 106, pp. 66–76, 2018.

[11] Y. Y. Zhan and R. C. Xu, "K-mean distance outlier factor detect for outlier pattern of time series," *Computer Engineering and Applications*, vol. 45, no. 9, pp. 141–145, 2009.

[12] M. Hossein, M. Sadegh, P. Alessandro, C. Ryad, and M. Vittorio, "Analyzing tracklets for the detection of abnormal crowd behavior," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pp. 148–155, Waikoloa, HI, USA, January 2015.

[13] M. Ajay, Z. H. Ge, J. Wang et al., "Rapid detection of maintenance induced changes in service performance," in *Proceedings of the Seventh Conference on emerging Networking Experiments and Technologies*, pp. 1–12, Tokyo, Japan, December 2011.

[14] Y. Zhang, Z. H. Ge, G. Albert, and R. Matthew, "Network anomography," in *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*, pp. 317–313, Berkeley, CA, USA, January 2005.

[15] J. Wong, C. Colburn, E. Meeks, and S. Vedaraman, "Rad—outlier detection on big data," *The Netflix Technology Blog*, vol. 19, 2015.

[16] A. Kejariwal, "Introducing practical and robust anomaly detection in a time series," Twitter Engineering Blog," *Web*, vol. 15, 2015.

[17] H. W. Xu, W. X. Chen, N. W. Zhao et al., "Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications," in *Proceedings of the 2018 World Wide Web Conference*, pp. 187–196, Lyon, France, April 2018.

[18] P. Malhotra, A. Ramakrishnan, G. Anand et al., "LSTM-based encoder-decoder for multi-sensor anomaly detection, ICML 2016 anomaly detection workshop," 2016, https://arxiv.org/abs/1607.00148.

[19] D. Li, D. C. Chen, L. Shi et al., "MAD-GAN: multivariate anomaly detection for time series data with generative adversarial networks," 2019, https://arxiv.org/abs/1901.04997.

[20] B. Zong, Q. Song, R. Q. Min et al., "Deep autoencoding Gaussian mixture model for unsupervised anomaly detection," in *the Proceedings of the International Conference on Learning Representations*, Vancouver, BC, Canada, February 2018.

[21] N. Gugulothu, P. Malhotra, L. Vig, and G. Shroff, "Sparse neural networks for anomaly detection in high-dimensional time series," in *AI4IOT Workshop in Conjunction with ICML*, International Joint Conference on Artificial Intelligence and European Conference on Artificial Intelligence, Stockholm, Sweden, 2018.

[22] Y. Su, Y. J. Zhao, C. H. Niu et al., "Robust anomaly detection for multivariate time series through stochastic recurrent neural network," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2828–2837, Anchorage, AK, USA, July 2019.

[23] P. Veličković, G. Cucurull, A. Casanova et al., "Graph attention networks," 2017, http://arXiv.org/abs/1710.10903.

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[25] R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu, "Exploring the limits of language modeling," 2016, http://arXiv.org/abs/1602.02410.

[26] K. Cho, B. V. Merrienboer, C. Gulcehre et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, http://arXiv.org/abs/1406.1078v3.

[27] C. Szegedy, W. Liu, Y. Q. Jia et al., "Going deeper with convolutions," 2014, http://arXiv.org/abs/1409.4842v1.

[28] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *In European Conference on Computer Vision*, vol. 1, pp. 818–833, 2014.

[29] P. Neill, D. Entekhabi, E. Njoku, and K. Kellogg, "The NASA soil moisture active passive (SMAP) mission: overview," in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium*, Ahmedabad, India, December 2010.