MTAD_RF: Multivariate Time-series Anomaly Detection based on Reconstruction and Forecast

Kenan Qin[1], Mengfan Xu[1], Bello Ahmad Muhammad[1,2], and Jing Han[1]
[1]School of Compter Science, Shaanxi Normal Unversity, Xi'an, ShaanXi, 710062, China
[2]Bayero University, Kano, Kano 700241, Nigeria

Anomaly detection in multivariate time series is an important research direction, which helps to improve the security of industrial systems by detecting abnormally unreliable devices. Multivariate time series (MTS) anomalies not only need to pay attention to the time correlation between different time series but also need to consider the abnormal changes in the relationship between different variables. Once the influence relationship between two variables that influence each other is ignored, it will likely lead to false positives or false positives. At the same time, the degree of influence between different time series or different features is also inconsistent, just like what happened recently have radically different influences on the present. Furthermore, most of the existing models are weak in detecting no abnormality. To tackle these issues, in this paper, we propose a new model of multivariate time series anomaly detection based on reconstruction and forecast, named MTAD_RF. First, we capture the temporal and feature correlations of MTS through two parallel GAT layers, and at the same time distinguish the influence degree between different time series or different features based on attention coefficients. Second, we leverage the generative power of VAE and the single-step forecast power of MLP to jointly detect known and unknown anomalies based on reconstructed and predicted models. Major practical implications of the proposed approach is missing. Finally, anomalies are detected and explained based on temporal and feature anomaly scores. Experiments demonstrate that our model outperforms current state-of-the-art methods on 4 real-world datasets, with an average F1 score of about 95% and excellent anomaly diagnostic ability.

Index Terms—**Anomaly Detection, Multivariate Time-series, Graph Attention Network,Variational AutoEncoder.**

I. Introduction

With the continuous development of the Internet of Things and 5G, the problem of data security becomes more and more prominent, especially the lack of reasonable detection mechanism in the security strategy, which may cause data theft, tampering or even destruction, posing a great threat to the security communication of data. Among them, the security problem of time series data is more severe due to the high requirements on the continuous working ability and accuracy of equipment. The most fundamental challenge in the security analysis of time series data is to mine the time correlation between data to detect the abnormal data timely and accurately. Accurate abnormal detection results can effectively guide relevant personnel to carry out accurate protection and protection of data to avoid huge economic losses caused by data anomalies. However, it is difficult to conduct anomaly detection research directly from massive and complex time series data, because of the diversity of data types, the large size of data and other noise interference. In recent years, the data analysis technology represented by deep learning has developed rapidly. It has become a hot research point to solve the problem of anomaly detection in complex data by using deep learning model. On the other hand, network science can comprehensively analyze complex relational data from multiple perspectives such as physics, computer and mathematics, and build different networks according to different data characteristics. This method can effectively focus more attention on data correlation and avoid the impact of noise generated by data acquisition on core data analysis. However, how to conduct proper network modeling for massive and complex time series data in real life and detect anomalies more quickly and accurately according to abundant data information has become an important task to be solved urgently.

Anomaly detection is used to find individuals or events that deviate from most node behaviors. It is widely used in various life and production scenarios, such as abnormal behavior monitoring of smart homes in the Internet of Things [1], sensor network device indicator abnormal monitoring [2], terminal abnormal signal detection in wireless body area network [3] and so on. To improve detection accuracy, it is particularly important to model more appropriately for real-world networks. Since sensor network devices are various and work continuously over time, they can be abstracted into multivariate time series (MTS) for anomaly detection [4] to detect which devices are abnormal at what time. As a result, figuring out how to effectively detect anomalies in multivariate time series has become an important challenge that has to be solved urgently.

Fortunately, there have been many studies on multivariate time series anomaly detection in the literature [5], [6], [7], [8] etc. For instance, in [9], [10], MTS is converted into univariate time series (UTS), and then the anomaly detection method of UTS is used. Although these methods present good approach for anomaly detection, the methods often ignore the relationship between different variables and achieved less performance results. [11], [12] regard MTS as a whole, first model the normal data behavior model according to the characteristics of

Manuscript received February 10, 2023; revised April 23, 2023. Corresponding author: Kenan Qin (qinkn@snnu.edu.cn)
Email Addresses: Mengfan Xu (cybersecurityxu@snnu.edu.cn), Bello Ahmad Muhammad (bamuhammad@snnu.edu.cn), Jing Han (hanjing@snnu.edu.cn).
This work is supported by the Natural Science Basic Research Plan in Shaanxi Province (2022JQ-594)

MTS, such as time correlation, then encode and reconstruct or predict the original MTS, and finally calculate the abnormal score according to the reconstruction or forecast error to judge the abnormal. Due to the high efficiency of deep learning, the use of various deep learning models to study MTAD is emerging recently.

However, there are still some shortcomings in the existing methods. LSTM_VAE [8] and OmniAnomaly [7] only consider the temporal correlation between different moments of MTS, ignoring the correlation between different variables, which is however crucial for detecting anomalies. For example, with sensors that measure voltage and current, etc., if one fails, the other will fluctuate accordingly. If we ignore the relationship between the two and focus only on the outlier changes of a single device indicator, it will often cause the normal fluctuations of the device to be regarded as abnormal, resulting in false positives or false positives. Additionally, the existing methods [13], [14] regard the importance of each time series or variable as the same, which is unreasonable. For example, the impact of the data of the previous 10 months on the current moment is almost negligible, and the importance of the hub node in the sensor network is higher than that of other ordinary nodes. In addition, an important content of anomaly detection is to detect those exceptions that have not occurred. This requires that the model should not only learn from historical data but also consider the randomness of variables (hidden space), which is not mentioned in most literature [15], [16]. On the other hand, methods based on reconstruction error or forecast error detect anomalies from the global and local perspectives, respectively. Relying on only one of the methods will inevitably lead to errors, and combining the two methods to find the optimal solution may improve the accuracy.

To address the constraints of the previous methods, we propose in this paper, a new framework named **MTAD_RF** a multivariate time series anomaly detection using reconstruction and forecast. The propose MTAD_RF treats the values of the variables in each time series and the value of the variable at different time as a whole, investigates the temporal and feature-based correlations among several time series, and extracts the relationships between nodes simultaneously.

In the proposed MTAD_RF, we first use two parallel attention layers to distinguish the important of different moments and different variable through weight coefficients. the feature attention is mainly use to extract the dependencies between different features, while the temporal attention focuses on changing the relationship of different time series. Secondly, we combine reconstruction model and forecast model to better represent the time series data. Overall, our contributions are as follows:

1) We propose a new self-supervised model to solve the multivariate time series anomaly detection problem unlike the previous approaches. The proposed MDAT_RF outperforms state-of-the-art methods on 3 real datasets, with an average F1-score improvement of 9%. The result indicates that Our method has good abnormality diagnosis ability and can be applied in various industrial systems.
2) We use two parallel attention layers(Feature attention and temporal attention) to simultaneously capture the temporal and feature correlations of multivariate time series data and do not require any prior knowledge.
3) We introduce a balance parameter to combine the advantages of reconstruction-based and forecast-based models. The reconstruction model learns a low-dimensional representation of time series data from a global perspective, while the forecast model only predicts the next timestamp data, and the joint optimization of the model is controlled by balancing parameters to find the best solution in global and local considerations.

The organization of the article is as follows. In section 2, we present some work related to this paper. In section 3, the problem and symbolic representation to be solved in this paper are defined. In section 4, we describe in detail our proposed multivariate time series data anomaly detection model MTAD_RF. In section 5, we compare our method with other baseline methods on 4 real datasets, and the results show that our proposed model is effective. In section 6, we mainly perform parameter analysis and anomaly diagnosis capability. In section 7, we summarize and look ahead to our work.

II. Related Works

At present, there are two main anomaly detection methods for multivariate time series: The first method is to use UTS detection method for each time series in MTS, which performs well in early research, but it ignores the time correlation between different time series. The second is to model the multivariate time series as a whole. Different from the previous classic anomaly detection methods KNN [17], PCA [18], SVM [19], and ARIMA [20], In recent years, scholars concentrate more and more on using deep learning methods [21] for anomaly detection using multivariate time series. Most of these approaches base their investigation on either the reconstruction-based method or the forecast-based method, and they all consider a variety of loss functions. We discuss some related researches that are linked to anomaly detection in multivariate time series based on each deep learning algorithm from the perspectives of reconstruction and forecast aspect.

A. Reconstruction-based Method

In the recent year, various approaches have been proposed in the literatures that uses reconstruction based methods for anomaly detection [11], [8], [7], [13], [16], [5]. This method learn the normal behavior pattens across time series data and detect anomalies by reconstructing errors in the original data. Autoencoder is typically representative of this type of method where the encoder compresses the input features and the decoder calculates the reconstruction error of the restored original data to judge the abnormality [11]. In order to adapt to the characteristics of time series data, Kieu et al. [12] proposed to add a sliding window to the autoencoder to further capture the time information of time series data. However, the relationship between time sequences in MTS is much more complicated than that in UTS, and its the most important problem that needs to be studied and overcome.

LSTM_VAE [8] replaces the encoder in the VAE model with LSTM to capture the temporal correlation of MTS, but does not consider the different degrees of influence between different time series, and the LSTM model is slow and computationally intensive. OmniAnomaly [7] uses VAE-based random module and GRU module to solve the temporal correlation and random variable problem of MTS, and detect anomalies based on the difference between the reconstruction probability and the threshold. However, the model does not explicitly propose a specific feature correlation method and does not consider the importance of different moments and different features.

USAD [13] uses the idea of Generative Adversarial Networks(GAN) to train two autoencoders to close the gap with the original distribution of the data, and then adversarially train the discriminator to enlarge the distribution in the input data distribution. The generator on the model simulates the error and improves the sensitivity of detecting anomalies, but it ignores the correlation between features and the importance of variables. The MSCRED model [14] introduces a convolutional encoder and an attention-based convLSTM to construct a feature matrix of inter-sensor correlation and temporal information, and gives the residual feature matrix to detect anomalies. However, it only captures the degree of hidden influence between different time series, and cannot capture the mutual influence between different sensors.

MTAD_GAT [16] creatively uses two parallel attention layers to capture temporal and feature correlations at the same time, and mines the hidden association importance of different time series and different features according to the attention weight coefficient. However, since the GRU cannot be calculated in parallel, with the gradual increase in the amount of input data and the size of the model, the calculation speed and amount of calculation will also increase significantly. TranAD [5] attempts to construct an adaptive combination of multiple meta-learning building blocks based on Transformer from a lightweight perspective, and amplifies the reconstruction error through two adversarial training processes to improve the speed and accuracy of detecting anomalies. We questioned that TranAD executes the multi-head attention mechanism serially, which is difficult to correlate time and features at the same time, and the adversarial training requires high model parameters, which may lead to non-convergence problems and make training difficult.

B. Forecast-based Method

In recent years, various methods for anomaly detection using forecast-based methods have also been proposed [10], [15], [22]. Similar to the principle of reconstruction model, the forecast model first performs feature extraction on MTS dimension reduction, and then predicts the next time series data according to the learned normal data model, and often takes the abnormal variable with larger forecast error as the abnormal variable, which has the advantage of considering the local area. The optimal solution is crucial for time series data, and the close connection between contexts makes this method efficient and feasible. [10] model each variable time series data separately, convert MTS to UTS and then input LSTM according to the forecast error. However, the model cannot effectively capture the interaction information between different variables, and this phenomenon is in the real network. unavoidable. The DeepAnt model [15] uses a window of previous observations as input to predict the next timestamp to build a CNN forecast model to detect outliers in multivariate time series. The DAGMM model [22] believes that the hidden information in the low-dimensional space helps to detect anomalies, and predicts the likelihood by feeding the compressed network information together into a Gaussian mixture model "GMM". However the input of the DAGMM is multivariate, but not a series of time series data. For multivariate time series data, time information is undoubtedly an important factor affecting the detection results.

Based on the limitations of existing methods mentioned above, there are still some shortcomings such as only considering the time correlation of MTS and ignoring the feature correlation, treating all-time series and the degree of mutual influence between variables without distinction, as well as the ability to detect those anomalies that have not occurred, that is, whether variable randomness is considered, and so on. Table I provide the summary of the existing approaches and shows the differences between the proposed MDAT_RF and the existing methods clearly. In this paper, we proposed a model that combine both the reconstruction and forecast method in order to improve the performance of detecting anomalies.

Table I. Differences Between Different Models

Item / Models	Temproal	Features	Importance	Stochasticity
DAGMM	✗	✔	✗	✔
OmniAnomaly	✔	✗	✗	✔
LSTM_VAE	✔	✗	✗	✗
USAD	✔	✗	✗	✔
MTAD_GAT	✔	✔	✔	✗
TranAD	✔	✗	✔	✗
MTAD_RF	✔	✔	✔	✔

Notes: Temproal means temproal correlation. Features means correlation of features. Importance means whether to consider the degree of influence between different time series. Stochasticity means whether to consider the detection of undiscovered anomalies

III. Preliminaries Formulation

Some prior knowledge is used to facilitate understanding of our proposed model. In this section, we first explain in details on how the proposed MTAD-RF can perform multivariate time series for anomaly detection, then we define the problem we want to addressed in this paper. Finally, the basic knowledge of GAT and VAE involved in MTAD_RF is introduced.

A. Anomaly Detection Structure

As shown in Figure 1, the Multivariate time series anomaly detection structure involves two stages of the assignment, which are $model\ training$ and $anomaly\ detection$.The data preprocessing module and relationship extraction module are shared by the model training and anomaly detection. The data preprocessing module, includes data cleaning, data normalization, and data partitioning by a sliding window. This

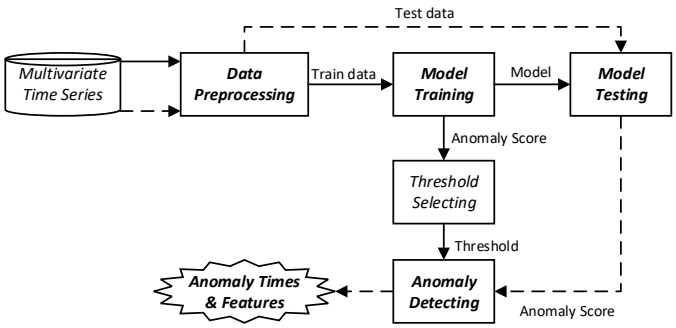


Figure 1. The structure of the MTAD_RF. The solid lines indicate the process of the training model. The dotted lines donate the process of anomaly detection.

module does not only eliminates the error effects of noise and inconsistent data types on the model but also avoids the invalid effects of outdated historical data. The relationship extraction module mainly captures the data characteristics of multivariate time series from the temporal correlation and feature correlation, which provides the basis for the subsequent accurate modeling of normal data behavior. The $model\ training$ is implemented mainly by the model training stage, which is a specific implementation of MTAD_RF. It repeatedly adjusts the parameters to train the model according to train data until it reaches the best state and saves it as the initial model of the model testing module. On the other hand, it transfers the anomaly score to the threshold selecting module to train the automatic selection of anomaly thresholds, laying the foundation for subsequent anomaly determination. The $anomaly\ detection$ is mainly for test data that contains abnormalities, relying on the model testing module to calculate the anomaly score at different times or features and anomaly thresholds, which send to the anomaly detecting module to get the anomaly times and features. In practical industrial systems, the model training phase is transparent, and only the anomaly detection phase is needed to detect anomalies.

B. Problem Definition

In order to facilitate modeling and express the problem we want to solve clearly, here we define the related problems in multivariate time series anomaly detection with a mathematical formal language.

Definition I (Multivariate Time Series): A multivariate time series $X_T = \{X_1, X_2, ..., X_i, ...X_t | i \in (1, t)\}$, where t indicates the maximum number of times. The i-th time series $X_i = \{X_{i1}, X_{i2}, ..., X_{ij}, ..., X_{ik} | j \in (1, k)\}$, where k indicates the number of features in X_i. Thus, $X_T \in \mathbb{R}^{t \times k}$.

Definition II (Multivariate Time Series Anomaly Detection): For a multivariate time series X_T, the purpose of anomaly detection is that judging whether the i-th time series X_i is anomalous by whether the corresponding time anomaly score S_i exceeds the threshold.

In particular, We assume that as long as there is at least one feature in X_i is anomalous, then X_i is an anomalous time series.

Definition III (Anomaly Interpretation): For an anomalous time series X_i, anomaly interpretation is to find out which anomalous features X_{ij} of the time series. By calculating feature anomaly score S_j, features whose anomaly score exceed the threshold are considered to anomalous features.

It should be noted that both the time anomaly score S_o and the feature anomaly score S_j are based on the anomaly score S_{ij} of the j-th feature at the i-th moment. We assume that the temporal anomaly score S_i of the time series X_i is the sum of the anomaly scores of all features at that moment, and the anomaly score S_j of each feature is the average of the scores of the corresponding features at all times.

C. Basics methods

In this section, we will briefly introduce the two more classic models GAT and VAE used in this paper.

1) Graph Attention Network (GAT)

Graph Attention Network (GAT) [23] was first proposed by Yoshua Bengio to solve problems that current Graph Convolution Networks (GCN) [24] cannot solve, including dynamic graph problems (especially when the data in the training and test sets are based on different graph structures), directed graph processing bottlenecks, and the problem of assigning different learned weights to different neighbors.

GAT focuses on obtaining the influence of other nodes on this node. GAT essentially has two operation modes, namely Mask graph attention or Global graph attention. Global graph attention, as the name implies, means each vertex i performs an attention operation on any vertex on the graph. The advantage is that it does not depend on the structure of the graph at all, and there is no pressure on inductive tasks. However, in Mask graph attention, the operation of the attention mechanism is only performed on the neighbor vertices.

GAT is mainly implemented in two steps: (1) Calculate the attention coefficient. For vertex i, calculate the similarity coefficient e_{ij} between its neighbors j and itself one by one, and then normalize it according to the softmax function. (2) According to the calculated attention coefficient, weight and aggregate the features. Output new features fused with neighborhood information for each vertex.

In essence, both GCN and GAT aggregate the features of neighboring vertices to the central vertex (an aggregate operation), and use the local stationery on the graph to learn new vertex feature expressions. The difference is that GCN uses the Laplacian matrix, and GAT uses the attention coefficient. To a certain extent, GAT will be stronger because the correlation between vertex features is better incorporated into the model.

2) Variational Auto-Encoder (VAE)

As a form of a deep generative model, Variational Auto-Encoders (VAE) is a generative network structure based on Variational Bayes (VB) inference proposed by Kingma et al.[25]. Unlike traditional autoencoders that describe the latent space numerically, it describes the observations of the latent space in a probabilistic way, showing great application value in data generation.

Table II. Notations Definitions

Notations	Definitions
X_T	multivariate time series
X_i	a time series in X_T
X_{ij}	a feature in X_i
V_j	a feature time series in X_T
$\hat{X}_T$	reconstructed X_T
$X_t{}'$	forecasted X_i
w	sliding window
S_i	a time series anomaly score
S_j	a feature time series anomaly score
S_{ij}	a feature in a time series anomaly score
h_f	feature attention result
h_t	temporal attention result
h	intermediate vector
z	hidden vector of VAE

Assuming that the original data is $X = \{x_i\}_i^N$, each data sample x_i is a randomly generated independent, continuous or discrete distribution variable. The generated data is $X' = \{x_i'\}_i^N$, and suppose that the process produces a latent variable Z, that is, Z is the mysterious cause (feature) that determines the properties of X. The observable variable X is a random vector in a high-dimensional space, and the unobservable variable Z is a random vector in a relatively low-dimensional space.

The generative model can be divided into two processes: (1) the approximate inference process of the posterior distribution of the latent variable Z: $q_\theta(z|x)$, which is an inference network. (2) the generating variable Conditional distribution generation process for X': $P_\theta(z)P_\theta(x'|z)$, which is generation network. The output of the inference network should be the posterior distribution $p(z|x)$ of Z. But this $p(z|x)$ posterior distribution itself is not easy to find. So some scholars came up with another scalable distribution $q_\theta(z|x)$ to approximate$p(z|x)$. By learning the parameters of $q_\theta(z|x)$ through a deep network, and optimizing q step by step to make it very similar to $p(z|x)$, it can be used to approximate the inference of complex distributions. To make the two distributions q and p as similar as possible, we can minimize the KL divergence between the two distributions.

IV. MTAD_RF Approach

In this section, we present a detailed description of the proposed MTAD_RF model. First, we present the overview of the model in Section 4.1. We then present the detail of each module of MTAD_RF model: Data preprocessing, relationship extraction, model training and anomaly detection and interpretation in section 4.2, 4.3, 4.4 and 4.5 respectively. Table II shows meaning of notations frequently used in this paper.

A. Overview of MTAD_RF

The overall framework of the MTAD_RF model is shown in Figure 2 below, which mainly includes four parts: Relationship extraction based on GAT, data reconstruction based on VAE, data forecast based on MLP, and anomaly detection and location. The specific functions of each part are as follows:

- *Relationship Extraction* : We apply a 1D convolution layer to the original data to enhance the local feature aggregation ability of the data when using sliding windows [26], while simplifying the computation. Then two analogous graph attention layers are applied to extract information from the original data from the perspectives of time and feature simultaneously. Finally, concatenate the three outputs as intermediate vector h.
- *Reconstruction* : Feed the middle vector h to the VAE to reconstruct the feature values at all times $\hat{W}_t$ by the sliding window w, where Z is recorded as a hidden vector in VAE.
- *Forecast* : A MLP with 3 layers is applied to the middle vector h to predict the feature values of the next moment X_t' according to the time series data in the sliding window w. The basis for forecast is the influence of adjacent time series data in continuous time.
- *Anomaly Detection* : First, calculating anomaly score of different features at different times S_{ij} according to the reconstruction error and forecast error at the t-th moment, and then get the final anomaly scores S_i and S_j of different times and different features. Therefore, the anomaly times are derived by S_i through compared anomaly threshold, and the abnormal features are obtained by combining the feature attention weight matrix and S_j.

B. Data Preprocessing

To reduce the interference of noisy data to the model, we first perform data preprocessing, mainly data cleaning and data standardization. Referring to the Spectral Residual (SR) algorithm [27] to process anomaly detection of univariate time series data, we use the SR method to clean the data; in addition, the data standardization adopts the maximum and minimum standardization method.

Data cleaning: Since the methods based on reconstruction and forecast are greatly affected by irregular values and outliers in the data, to reduce the influence of data on the model method, we follow the most outstanding method for univariate data anomaly detection proposed by Ren et al.[27] Method SR method, which is used to detect anomalies for each moment of the training data. In addition, referring to the practice of [16], we replace the data at abnormal times with data that is close to normal to complete the data cleaning.

Data standardization: We use min-max normalization to process the training data, which is implemented as follows:

$$\tilde{V_j} = \frac{V_j - min(V_j)}{max(V_j) - min(V_j)} \tag{1}$$

where $V_j(j \in (1,k))$ is the j-th feature of all times, $max(V_j)$ is the maximum value of V_j, and $min(V_j)$ is the minimum value of V_j.

C. Relationship Extraction

The distinguishing feature of time series is long-term coherence, however, the impact of data from a long time ago on the current is often small or even irrelevant. Therefore, we

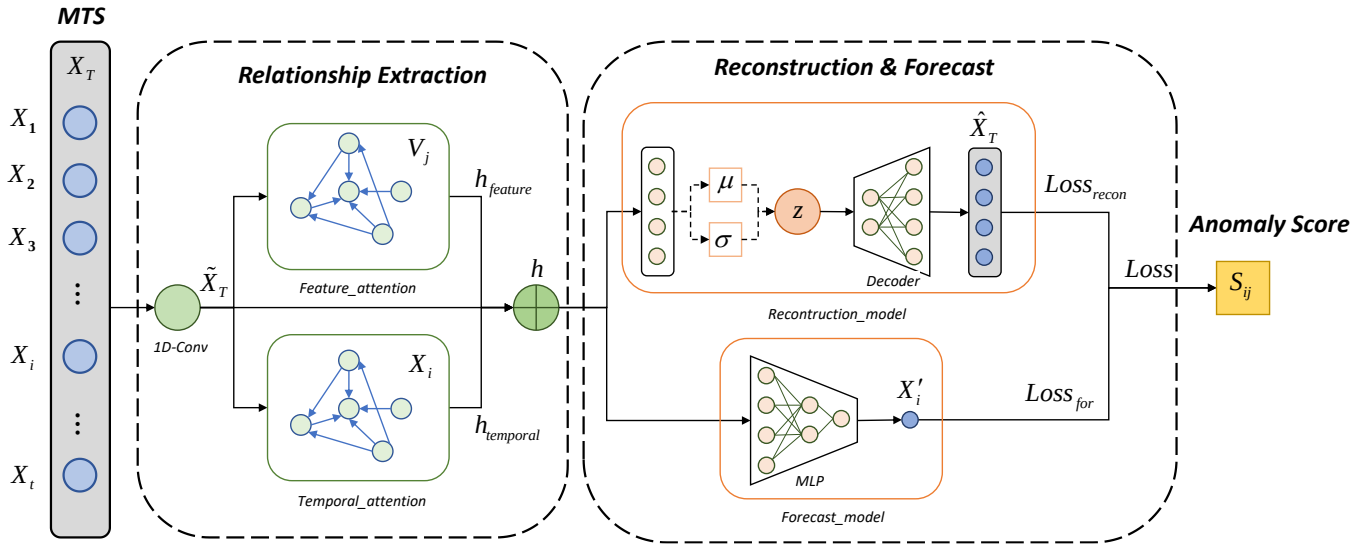


Figure 2. The framework of the MTAD_RF. The size of X_T is $n \times k$, h is the intermediate vector, which is the splicing of the results of 1D-Conv, $h_{feature}$ and $h_{temporal}$, size is $n \times 3k$. μ and σ represent the mean and variance, respectively. z is the encoded hidden vector. S_{ij} is the anomaly score of the j-th feature at the i-th time.

introduce the concept of *sliding window*. As the name implies, its main function is to divide too long time data into a window, to focus on the influence of local time series data on each other. As mentioned in [27], the convolution operation can work well for local feature extraction. Therefore, we preliminarily apply a 1D convolution layer to the original data X_T to initially focus on the interaction between local sequences, and the output is $\tilde{X_T} \in \mathbb{R}^{t \times k}$.

Additionally, time correlation and feature correlation of multivariate time series contains a lot of information, especially the mutual influence between features, which is very important for analyzing anomalies. GAT [23] can mine the mutual influence relationship between different nodes in the graph data by constructing the attention weight matrix. Therefore, we used a two-layer GAT layer to extract the feature information of time series data from the aspects of time and feature, respectively. The main principle of GAT is shown in the Figure 3.

For a network of n nodes $G = \{z_1, z_2, ..., z_n\}, z_i \in R^k$, k is the number of features of the node. To obtain a low-dimensional representation of the network with more information, all nodes share a weight matrix $W \in R^{k \times k'}$, and then perform self-attention on the nodes, that is, a shared attention mechanism $\mathbf{a}$, and calculate the attention coefficient accordingly:

$$e_{ij} = a(W\vec{z_i}, W\vec{z_j}) \tag{2}$$

which represents the influence of node j on node i, where $j \in N_i$. N_i is the set of first-hop neighbor nodes of node i. The attention mechanism $\mathbf{a}$ is implemented as follows:

$$e_{ij} = LeakyReLu(\vec{a}^T[W\vec{z_i} \parallel W\vec{z_j}]) \tag{3}$$

The weight vector $\vec{a} \in R^{2k'}$, $\parallel$ represents the splicing operation, $.^T$ represents the transpose operation, and LeakyReLu is

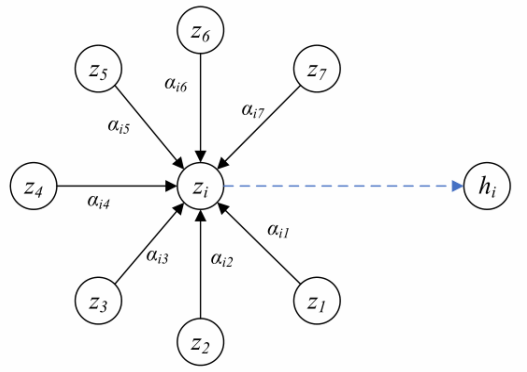


Figure 3. Attention mechanism between node z_i and its neighbors $z_{j,j\in(1,7)}$, $\alpha_{ij,j\in(1,7)}$ represent the corresponding attention coefficients. h_i is the output feature vector of z_i.

the activation function. In addition, to facilitate the comparison of different node attention coefficients, regularizing all j with the Softmax function:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{N_i} \exp(e_{ik})} \tag{4}$$

Finally, the feature vector h_i of node i is output, where $h_i \in R^{k'}$, σ is the activation function.

$$h_i = \sigma\left(\sum_{j \in N_i} \alpha_{ij} W\vec{z_j}\right) \tag{5}$$

Since the structural relationship between the variables of multivariate time series is not clear in reality, we propose to construct time series data as a fully connected graph from the perspective of time and feature, and then apply two-parallel GAT layers to extract information. Subsequent experimental results also demonstrate the effectiveness of our method.

1) Feature_Attention

From the perspective of feature, we regard multivariate time series as a graph $G_f = \{V_1, V_2, V_3..., V_k\}$, where node $V_j (j \in \{1, k\})$ has t features. It also can be understood in Figure 3. that we define one feature as a node and all-time feature values are node features. After sliding window w, the $\tilde{X_T}$ divided by multiple small pieces of size $R^{k \times w}$, which can acquire a series of a smaller graph G_w like the organization of G_f. In every G_w, we use GAT to calculate the association between two features, resulting in the attention coefficient α_f and the output $h_{feature}$, which size is $k \times t$.

2) Temporal_Attention

Similarly, from the perspective of time, we regard multivariate time series as a graph $G_t = \{X_1, X_2, X_3..., X_t\}$, where node $X_i (i \in \{1, t\})$ has k features. Different from feature attention, we define one moment as a node, and all feature values at this moment are node features. The $\tilde{W_t}$ is divided into a series of small pieces of size $R^{w \times k}$ according to the sliding window w. In the same way, GAT is applied in the sliding window to extract the relationship between two-time series, and obtain the attention coefficient α_t and the final output $h_{temporal}$, which size is $t \times k$. More implementation details can be seen in the Algorithm 1.

D. Model Traning

After initially extracting the time and feature features of the original data X_T, the next step is to train the model according to the minimization of reconstruction error and forecast error, which mainly includes the following parts.

1) Reconstruction_model

After extracting the features of the original data X_T, we concatenate $\tilde{X_T}$, $h_{feature}$ and $h_{temporal}$ as intermediate vector h. Since the difference between normal data is not large after encoding and decoding, the difference between abnormal data is obvious. Therefore, the reconstruction model mainly judges the abnormality according to the reconstruction error. VAE [25] has a stronger generation ability than general AE, and the generalization effect is excellent when reconstructing the original data. The schematic diagram of the specific VAE is shown in Figure 4.

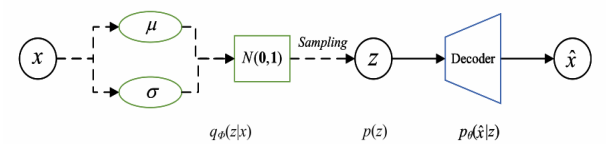


Figure 4. Schematic diagram of the principle of VAE.

The main idea of VAE is to use some common distribution like a normal distribution train a model $x = G(z)$ that maps the original simple probability distribution to the true probability distribution of the training set. Here, the latent variable z is introduced, which is generated by the sampling and parameters of the input data x, and not only contains the information of x, but also satisfies the normal distribution.

First, for the continuous variable z according to the total probability formula, we can calculate $p(x)$:

$$p(x) = \int_z p(x|z) \cdot p(z) \, \mathrm{d}z \tag{6}$$

But it is difficult to calculate an integral part directly because we can't enumerate all the vectors z and $p(x)$ we don't know. The posterior distribution can be given by

$$p(z|x) = p(z) \cdot p(x|z)/p(x) \tag{7}$$

It is equally hard to calculate directly. Therefore, we introduce a inference model $q_\phi(z|x)$ and a generative model(decoder) $p_\theta(\hat{x}|z)$ to deal the problems, which is shown in Figure 4. So, the specific implementation of the reconstruction model is the followings, note that ϕ, θ are all parameters that need to be trained.

- *Encoding* : it is assumed that the latent variables z follow a normal distribution $q_\phi(z|x)$ with respect to each sample x posterior, so that the z collected by $p(z)$ uniquely corresponds to a certain sample; and then based on the posterior $q_\phi(z|x)$ sampling to obtain $p(z)$. During training, each sample trains two targets mean μ and variance σ.
- *Decoding* : Calculate $p_\theta(x|z)$ to regenerate sample $\hat{x}$.

To make all posterior distributions align with the standard distribution, KL divergence is introduced to achieve. And combined with the reconstruction error, the loss function of the reconstructed model can be given by:

$$\begin{aligned} Loss_{recon} &= \left\| \hat{W}_t - W_t \right\|_2 + KL\left(N(\mu, \sigma^2) || N(0,1)\right) \\ &= \sum_{i=1}^{t} \left\| \hat{X}_i - X_i \right\|_2 + \frac{1}{2}\left(-\log\sigma^2 + \mu^2 + \sigma^2 - 1\right) \end{aligned} \tag{8}$$

The KL divergence loss is a regular term, which makes the encoding result have zero mean and a certain amount of noise, which increases the generalization ability of the decoded and regenerated samples.

2) Forecast_model

The intermediate vector h is passed into the Multi-Layer Perceptron(MLP) [28] , through the multi-layer hidden layer, and finally, the t-th time data predicted at the t-h moment X'_t is output. The loss function adopts MSE(mean-square error):

$$Loss_{for} = \left\| X'_t - X_t \right\|_2 \tag{9}$$

3) Joint optimization

Unlike most models that only consider reconstruction or forecast in a certain way to detect anomalies, we believe that the joint reconstruction and forecast model can comprehensively detect model robustness for accurately judging whether the i-th moment is abnormal or not. At the same time, we set hyperparameters $\eta \in \{0, 1\}$ to balance the effects of reconstruction and forecast models. Therefore, the total loss function of the model is as follows:

$$Loss = \eta Loss_{recon} + (1 - \eta) Loss_{for} \tag{10}$$

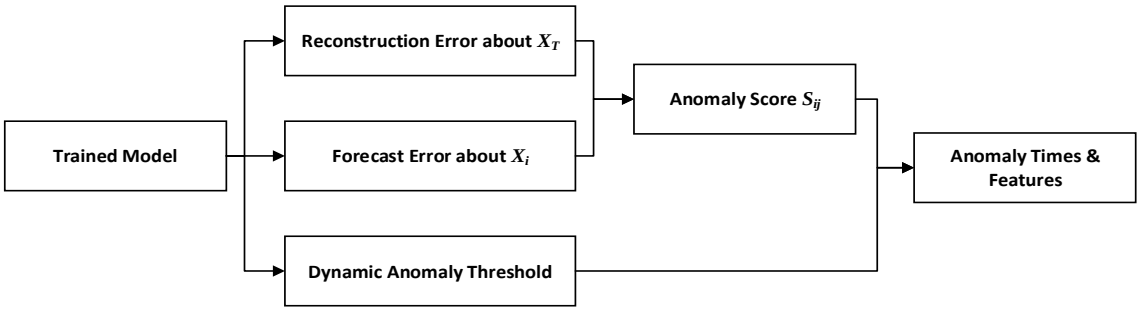


Figure 5. The process of detecting MTS anomalies from trained model by anomaly score.

E. Anomaly Detection and Interpretation

The main process of MTS anomaly detection according to the trained model is shown in Figure 5. First of all, the trained model can generate all reconstructed time series $\hat{X_T} = \{\hat{X_1}, \hat{X_2}, ..., \hat{X_t} | i \in (1, t)\}$, a predieted time series $X_i{}' = \{X_{i1}{}', X_{i2}{}', ..., X'_{ik} | j \in (1, k)\}$and dynamically generate the abnormal threshold according to the training data, where the abnormal threshold determination method comes from ϵ-method proposed by [10]. Then, the abnormal time and feature can be determined by calculating the abnormal score S_{ij} of the j-th feature at the i-th time and comparing it with the threshold.

Referring to the definition of anomaly detection defined above, we introduce the parameter $\lambda \in \{0, 1\}$ to balance the reconstruction error of X_T and the forecast error of X_i to calculate the anomaly score S_{ij} of the j-th feature at the i-th time:

$$S_{ij} = \left[\lambda \left\| \hat{X_{ij}} - X_{ij} \right\|_2 + (1 - \lambda) \left\| X'_{ij} - X_{ij} \right\|_2\right] \quad (11)$$

We detect anomaly on two aspects *anomaly times* and *anomaly features*, which corresponds to the results of anomaly detection and anomaly interpretation. We define temporal anomaly score $S_t = \{S_1, S_2, ..., S_i, ..., S_t | i \in (1, t)\}$ as the sum of the anomaly scores for all features at that moment, feature anomaly score $S_f = \{S_1, S_2, ..., S_j, ..., S_k | j \in (1, k)\}$ is the average of all moments for an feature, which is computed as follows.

$$S_i = \sum_{j=1}^{k} S_{ij}, S_j = \frac{1}{t} \sum_{i=1}^{t} S_{ij} \quad (12)$$

Once a certain time is judged to be an abnormal time, the features that exceed the feature abnormal score threshold at that time are immediately obtained as candidates for abnormal causes. The entire process from model training to anomaly detection is specifically implemented as shown in algorithm 1. In our approach, GAT can be used to learn the temporal and feature correlations of MTS and assign their weights according to the form of input data. The anomaly score result is used to judge whether the next moment is abnormal and locate the abnormal features.

Remark: The time complexity of algorithm 1 is $O(max(t^2, k^2))$, and the space complexity is $O(max(t, k))$. In the process of algorithm execution, MTS adaptively learns the smallest possible constraints and an attention strategy that can extract as much information as possible. After the epoch is executed, the abnormal score values of all time series and all features of MTS can be obtained, which meets the expectation of abnormal judgment. In addition, since each time series interacts with data in adjacent sliding window of size w, it is parallel in each iteration, indicating that the algorithm is fair, which avoids that different time sequences are affected by the previous extraction results due to the sequence of occurrence.

V. Experiments and Results

To demonstrate the effectiveness and accuracy of the proposed MTAD_RF model, we conducted the following experiments. Unfortunately, it include comparing the performance of the proposed MTAD_RF on multiple real world datasets and multiple baseline methods, as well as an analysis of important parameters in our model and anomaly detection diagnostics of our model. We first describe the datasets, the baseline methods and the evaluation metrics for performance evaluation.

A. Datasets

We selected 4 publicly available datasets to test our model. SMD (Sever Machine Dataset) is a 5-week-long dataset collected from a large internet company with policy permission [7]. SMAP (Soil Moisture Active and Passive) and MSL (Mars Science Laboratory rover) are satellite datasets collected by NASA [29]. SWaT (Secure Water Treatment) is a water treatment test bench for research in the field of cybersecurity [30]. The specific information of the dataset is shown in Table III.

B. Baseline Methods

To demonstrate the overall performance of our proposed method MTAD_RF, we compare it with 5 state-of-the-art baseline methods. The details are as follows:

- AutoEncoder [11]: Encoders and decoders that unsupervised learn normal data patterns in multivariate time series data and detect anomalies through reconstruction errors.
- OmniAnomaly [7]: A Stochastic Recurrent Neural Network that stitches together VAE and GRU, taking into account both time dependence and randomness. Random variables can capture more information from historical random variables and better represent the input data.

Algorithm 1: The Learning Algorithm of MTAD

Input: Multivariate time seres$X_T \in \mathbb{R}^{t\times k}$, Epoch T, Sliding window w, Balance parameter λ and η.
Output: Anomaly temporal and feature score S_t and S_f.

Clean and standardize X_T;
Divide X_T by slide window w;
$\mathbf{Loss_r} \leftarrow 0$, $\mathbf{Loss_f} \leftarrow 0$, $\mathbf{Loss} \leftarrow 0$;
for *epoch* $\leftarrow 1$ *to* T **do**
  $\tilde{X_T} \leftarrow$1D-Conv(X_T);
  // Temporal Attention return $\mathbf{h}_{temporal}$
  foreach X_i *in* $\tilde{X_T}$ **do**
    foreach X_j *in* $\tilde{X_T}$ **do**
      $\mathbf{e}_{ij} \leftarrow LeakyReLu(\vec{\mathrm{a}}^T[\mathbf{W_1}X_i \parallel \mathbf{W_1}X_j])$;
      $\alpha_{ij} \leftarrow Softmax(e_{ij})$;
      $\mathbf{h}_i \leftarrow \sigma\left(\sum_{j\in T}\alpha_{ij}\mathbf{W_1}X_j\right)$;
    end
  end
  // Feature Attention return $\mathbf{h}_{feature}$
  Transpose $X_T{}'$, do the above loop over each row of V_i;
  $\mathbf{h} \leftarrow \tilde{X_T} \parallel \mathbf{h}_{temporal} \parallel \mathbf{h}_{feature}$;
  // Reconstruction Model
  $\mu \leftarrow \mathbf{W_2}\mathbf{h}$;
  $\sigma_i \leftarrow \mathbf{W_3}\mathbf{h}$;
  Sample ϵ from $N(0,1)$;
  $z \leftarrow \mu + \epsilon * \sigma$;
  $\hat{X_T} \leftarrow \mathbf{W_4}z$;
  $\mathbf{Loss_{recon}} \leftarrow \left\|\hat{X_T} - X_T\right\|_2 + KL\left(N(\mu,\sigma^2)||N(0,1)\right)$;
  // Forecast Model
  $X_t{}' \leftarrow \mathbf{W_5}\mathbf{h}$;
  $\mathbf{Loss_{for}} \leftarrow \|X_i' - X_i\|_2$;
  $\mathbf{Loss} \leftarrow \eta Loss_{recon} + (1-\eta)Loss_{for}$;
  Update $\mathbf{W_1}$,$\mathbf{W_2}$,$\mathbf{W_3}$,$\mathbf{W_4}$,$\mathbf{W_5}$, with $\mathbf{Loss}$;
  if *epoch* = T **then**
    $S_t \leftarrow \lambda||\hat{X_T} - X_T||_2 + (1-\lambda)\|X_t' - X_t\|_2$;
    $S_f \leftarrow \frac{1}{T}[\lambda||\hat{X_T} - X_T||_2^{\mathbf{T}} + (1-\lambda)\|X_t' - X_t\|_2^{\mathbf{T}}]$;
  end
end
Return S_t and S_f.

Table III. Details of each dataset

Datasets	Features	Train	Test	Anomalies
SMD	38	708405	708420	4.16%
SMAP	25	135183	427617	13.13%
MSL	55	58317	73729	10.72%
SWaT	51	495000	449919	12.14%

Table IV. Parameter settings in Experiments

Parameters	Default Value
batch size	256
window size	50
training iterations	30
optimizer	Adam
learning rate	0.001
valid split	0.1
Decoder layers	1
MLP layers	3

However, it does not explicitly solve the problem of feature correlation in the model and ignores the interactive information between features.

- LSTM_VAE [8]: A reconstructed model composed of LSTM and VAE. The LSTM module is used to capture time information, and the VAE module captures feature information, but the LSTM module takes more time.
- DAGMM [22]: Anomaly detection is performed through a neural network model based on Autoencoder and Gaussian Mixture Model "GMM" that comprehensively considers hidden layer features and reconstruction errors, but it ignores the time information of the data.
- USAD [13]: An unsupervised anomaly detection method based on an adversarially trained autoencoder is proposed. The use of adversarial training and its architecture enables it to isolate anomalies while providing fast training. It also demonstrates its superiority in robustness, training speed, and high anomaly detection.

C. Experiment Setup

The hardware environment of our experiment depends on the Linux system, with 8G memory, 4 CPU cores, and 1 integrated graphics card GeForce RTX3090; the software environment is pytorch1.10, cuda10.3. The main parameter values involved in the model and default value settings are shown in Table IV.

In addition, due to the large difference in features between datasets, the dimensions of each hidden layer in the reconstruction and forecast models of each dataset are different. The specific details are that for SMD, SMAP, MSL, and SWaT, the dimensions of the encoding part of the hidden layer in the reconstruction module are 100, 150, 100, 150; the dimensions of the decoding part of the hidden layer are 30, 50, 30, 50; the dimensions of the MLP hidden layer in the forecast module are 10, 30, 10, 30.

D. Experiment Results and Analysis

We experimentally investigate the anomaly detection performance of MTAD_RF and other baseline methods on 4 real datasets. The accuracy of anomaly detection of each method is compared, and the performance of our method is evaluated in two aspects: parametric analysis and anomaly diagnosis ability.

1) Performance of Accuracy

We use precision(P), recall(R), and F1-score(F1) as the evaluation metrics to test the performance of our model, which are calculated as follows. The reason we do not use the

accuracy(ACC) is that in the case of imbalanced positive and negative samples, the accuracy rate is a big flaw. For example, in anomaly detection, anomalies are often very rare, generally only a few thousandths. Therefore, even if all forecasts are negative (normal samples), the ACC is more than 99%, which is meaningless.

$$P = \frac{TP}{TP + FP}, \qquad R = \frac{TP}{TP + FN}, \qquad F1 = 2 \cdot \frac{P \cdot R}{P + R}$$

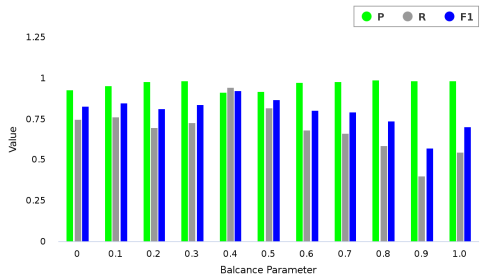
Where TP represents the number of positive samples that are correctly identified, FP represents the number of false negative samples, and FN represents the number of false positive samples.

We compare MTAD_RF with other baseline methods on the following 4 real-world datasets with precision (P), recall (R), and F1-score (F1) as evaluation metrics The specific experimental results obtained are shown in Table V.

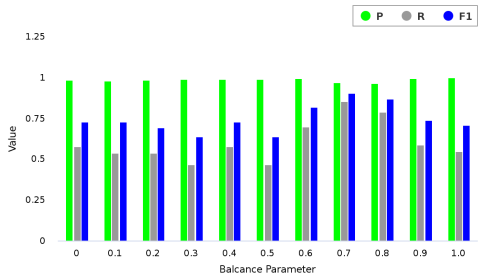
As can be seen from Table V, we can find that MTAD_-RF outperforms other methods on SMAP, MSL and SWaT datasets, and is second only to the USAD method on SMD datasets. The experimental results fully demonstrate the effectiveness and efficiency of our method. We also found that the AE method unexpectedly performed evenly across datasets despite its simpler structure, suggesting a lack of model personalization. At the same time, the OmniAnomaly method has lower performance than our method because it does not explicitly solve the feature-related problems in the model and ignores the interaction information between features. The DAGMM method is slightly inferior to other methods on several datasets, mainly due to the shortcomings of the algorithm design itself, which does not consider the time feature and only uses the feature feature. Although the LSTM-VAE method increases the capture of temporal features in VAE, the LSTM module is limited by long-distance "memory", and the parallel computing overhead is huge, so its experimental effect is not ideal. In addition, although the USAD method draws on the idea of GAN and trains two autoencoders adversarially and obtains a certain good implementation effect, the experimental complexity is high (a large number of parameters) and it is easy to cause non-convergence problems, so the effect is lower than ours method.

2) Parameters Analysis

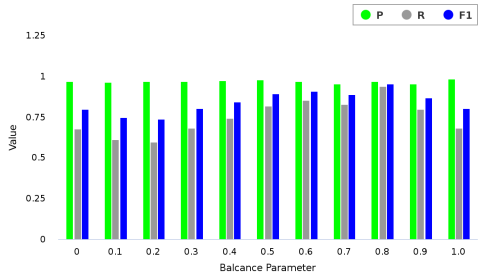
We discuss the joint optimization effect of the reconstruction and forecast modules on the model. In MTAD_RF, we propose the parameter η in the total loss function to balance the reconstruction loss and the forecast loss to minimize the error between the calculated result and the original data. Similarly, we also introduce the parameter λ in the anomaly score calculation to balance the effects of reconstruction error and forecast error on the calculated score. These two parameters work are samely, and we naturally keep them consistent in our experiments. To further study the specific impact of the reconstruction and forecast modules on the experimental results, we conducted additional experiments with different values of λ and η at an interval of 0.1 between [0,1] on SMD, SMAP, and MSL datasets about Precision, Recall and F1-score.



(a) parameter analysis of MSL



(b) parameter analysis of SMD



(c) parameter analysis of SMAP

Figure 6. The effect of different balance parameter λ and η in the MTAD_RF of 3 different datasets.

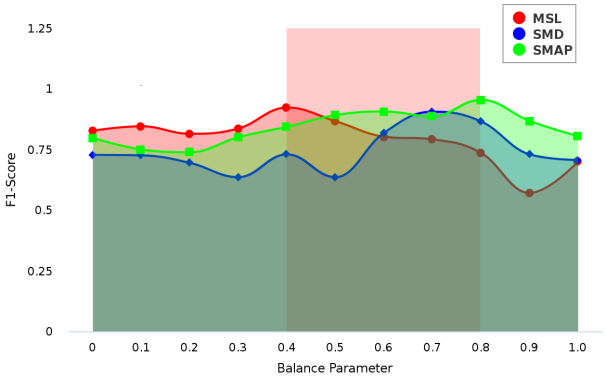


Figure 7. Comparison of F1-Score of 3 datasets with different balance parameters λ and η.

The results are detailed in Table VI and Figure 6 shows the different parameter analysis of three datasets. We notice that different settings of η and λ have similar effects on the three indicators. Precision has high values on the three datasets, and does not change greatly with the change of the balance parameter, while the Recall rate changes drastically. Therefore, F1-score is more suitable as an indicator to evaluate

Table V. Performance Comparison Results of methods on datasets.

Methods	SMD			SMAP			MSL			SWaT		
	P(Precision)	R(Recall)	F1	P	R	F1	P	R	F1	P	R	F1
AE	0.883	0.804	0.828	0.722	0.98	0.778	0.854	0.975	0.879	0.991	0.704	0.823
OmniAnomaly	0.764	0.974	0.856	0.647	0.995	0.784	0.772	0.971	0.86	0.963	0.745	0.82
DAGMM	0.789	0.917	0.848	0.699	0.884	0.781	0.723	0.696	0.709	0.895	0.734	0.806
LSTM_VAE	0.951	0.859	0.903	0.726	0.821	0.77	0.835	0.763	0.797	0.946	0.78	0.855
USAD	0.932	0.962	**0.938**	0.77	0.983	0.819	0.881	0.979	0.911	0.987	0.74	0.846
MTAD_RF	0.967	0.852	0.906	0.971	0.942	**0.956**	0.913	0.946	**0.923**	0.95	1.0	**0.974**

Table VI. The effect of different parameters on the MTAD_RF model on different datasets

λ&η	MSL			SMD			SMAP		
	P	R	F1	P	R	F1	P	R	F1
0	0.930	0.746	0.828	0.984	0.577	0.728	0.971	0.677	0.798
0.1	0.954	0.761	0.846	0.978	0.537	0.726	0.963	0.614	0.750
0.2	0.977	0.699	0.815	0.986	0.537	0.695	0.969	0.597	0.739
0.3	0.983	0.729	0.837	0.988	0.469	0.636	0.967	0.683	0.801
0.4	**0.913**	**0.946**	**0.923**	0.991	0.577	0.730	0.976	0.744	0.844
0.5	0.92	0.82	0.867	0.988	0.469	0.636	0.981	0.817	0.892
0.6	0.976	0.682	0.803	0.992	0.697	0.819	0.968	0.853	0.907
0.7	0.981	0.665	0.792	**0.967**	**0.852**	**0.906**	0.953	0.829	0.887
0.8	0.991	0.586	0.736	0.964	0.786	0.866	**0.971**	**0.942**	**0.956**
0.9	0.984	0.403	0.571	0.994	0.589	0.739	0.954	0.796	0.868
1.0	0.982	0.545	0.701	0.997	0.548	0.707	0.983	0.682	0.805

the performance of the model. We separately plotted the F1-scores of the three datasets under different balance parameters as shown in Figure 6. In addition, with the change of balance parameters, the F1-score of MTAD_RF is better than relying only on reconstruction (λ, η = 1) or forecast model (λ, η = 0) in most cases, which effectively proves that we introduce correctness of balancing parameters.

As shown in Figure 7, a parameter of 0 means only the forecast model, and a parameter of 1 means only the reconstruction model. We notice that the best F1 values are not at the ends of the polyline, but somewhere in the middle: for MSL is η and λ = 0.4, for SMD when η and λ = 0.7, for SMAP when η and λ = 0.8. The experimental results are in line with expectations and prove our idea that comprehensive reconstruction and forecast models are more efficient than relying only on one or the other. high. It's just that the balance ratio of reconstruction and forecast models for different datasets is different, which is also very reasonable, mainly related to the characteristics of the dataset. Secondly, for different datasets, we observed that when η and λ are within a certain interval, such as MSL when η and λ belong to [0, 0.6], the experimental results of MTAD_RF are better This demonstrates the robustness of our parameters compared to most baseline methods.

3) Anomaly Diagnosis Analysis

For real industrial systems, it is not enough to only detect the occurrence of anomalies, and it is also necessary to determine the specific causes of the anomalies, such as certain sensors that may be attacked. Our model achieves this function well. According to Equations 11 and 12, we can get a series of feature anomaly score S_j and temporal anomaly score S_i. By sorting the anomaly scores, we consider S_j and S_i which exceeding thershold are anomalies.

Since the datasets MSL, SMAP, and SMD record the time of anomaly generation and the specific features that caused the anomaly to occur, we choose them to test the anomaly

Table VII. Model diagnostic capability

Model	Datasets	HitRate@100%	HitRate@150%	NDCG@5
MTAD_RF	SMD	0.834	0.852	0.994
	SMAP	0.827	0.843	0.986
	MSL	0.805	0.827	0.980

diagnostic ability of our model. We select the top-8 features as the cause of the anomaly. We achieve our goal based on the metrics $HitRate@P\%$[7] and $NDCG$[31]. For

$$HitRate@P\% = \frac{Hit@\lfloor P\% \times |GT_t| \rfloor}{|GT_t|}$$

in which GT_t is the ground truth array of features that caused the anomaly moment, $|GT_t|$ is the length of GT_t, AS_t is the feature anomaly score, and $Hit@P\%$ indicates the ratio of the number of overlapping features between $top\lfloor P\% \times |GT_t| \rfloor$ and GT_t in AS_t to $|GT_t|$. Unlike $HitRate@P\%$ that considers the importance of each feature of time series data to be the same, Normalized Discounted Cumulative Gain (NDCG) considers the ordering factor, so that the feature with a higher anomaly score has a higher gain, which is calculated as

$$NDCG@k = \frac{DCG}{IDCG}$$

About $DCG = \sum_{i=1}^{k} \frac{rel(i)}{log_2(i+1)}$, which $rel(i)$ represents the correlation coefficient of i, and $IDCG$ means $idea\ DCG$, which calculate the best DCG according to the descending order of $rel(i)$.

We set P as 100 and 150 and k as 5 with three datasets, respectively, and the results are shown in Table VII. The experimental results confirm that our model can well find the top feature that causes anomalies. 80% of the true anomaly features are captured in all three datasets, and almost all of the top 5 anomalies in the ranking are detected, further implying the high performance of our method in finding the top 5 anomalies generated by anomalies reasons. For real industrial systems, our algorithms can greatly help users find the real cause of anomalies and solve problems as soon as possible.

VI. Conclusion

In this paper, we propose MDAT_RF for multivariate time series anomaly detection to address the problems of existing anomaly detection methods. For example, the time correlation and feature correlation between different time series are not considered at the same time, the degree of influence on different time series and variables is not distinguished, and the anomalies that have not occurred cannot be detected. In

MTAD_RF, we first use two parallel GAT layers named temporal attention and feature attention to simultaneously capture the temporal and feature correlations of MTS, and distinguish the influence degree between different time series and features according to the attention weight coefficient. Second, we combine the VAE-based reconstruction model and the MLP-based forecast model to detect those known and unknown anomalies. We validate our method is effective and efficient on 4 real datasets, yielding an average F1-score of 95%, an average improvement of 6% over the best baseline methods. Moreover, the abnormal diagnostic ability of our method performs well in HitRate and NDCG. From the perspective of actual production and life, our proposed MTAD_RF can enable industrial complex systems to automatically and accurately detect abnormal moments and equipment, and improve the security and safety of the system.

Nevertheless, our work still has some limitations, such as the initial assumption of a fully connected relationship between different time series and different features when learning attention coefficients. Future work in this area can investigate the use of various graph representation learning techniques (such as LINE [32], SDNE [33] and EGES [34] etc.) for anomaly detection to construct combinations between MTS features and further improve the feature extraction capability of MTS.

REFERENCES

[1] G. Sivapalan, K. K. Nundy, S. Dev, B. Cardiff, and D. John, "Annet: A lightweight neural network for ECG anomaly detection in IOT edge sensors," IEEE Transactions on Biomedical Circuits and Systems, vol. 16, no. 1, pp. 24–35, 2022.

[2] M. Jain, G. Kaur, and V. Saxena, "A K-means clustering and SVM based hybrid concept drift detection technique for network anomaly detection," Expert Systems with Applications, vol. 193, p. 116510, 2022.

[3] S. G S and R. Balakrishnan, "A statistical-based light-weight anomaly detection framework for Wireless Body Area Networks," The Computer Journal, vol. 65, no. 7, pp. 1752–1759, 2021.

[4] R. Chalapathy and S. Chawla, "Deep Learning for Anomaly Detection: A Survey," arXiv.org, 2019, [Online]. Available: https://arxiv.org/abs/1901.03407.

[5] S. Tuli, G. Casale, and N. R. Jennings, "Tranad," Proceedings of the VLDB Endowment, vol. 15, no. 6, pp. 1201–1214, 2022.

[6] X. Wang, D. Pi, X. Zhang, H. Liu, and C. Guo, "Variational transformer-based anomaly detection approach for multivariate time series," Measurement, vol. 191, p. 110791, 2022.

[7] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, "Robust anomaly detection for multivariate time series through Stochastic Recurrent Neural Network," Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019.

[8] D. Park, Y. Hoshi, and C. C. Kemp, "A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder," IEEE Robotics and Automation Letters, vol. 3, no. 3, pp. 1544–1551, 2018.

[9] J. Li, W. Pedrycz, and I. Jamal, "Multivariate Time Series Anomaly Detection: A framework of Hidden Markov models," Applied Soft Computing, vol. 60, pp. 229–240, 2017.

[10] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using lstms and Nonparametric dynamic Thresholding," Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018.

[11] Z. Chen, C. K. Yeo, B. S. Lee, and C. T. Lau, "Autoencoder-based network anomaly detection," 2018 Wireless Telecommunications Symposium (WTS), 2018.

[12] T. Kieu, B. Yang, and C. S. Jensen, "Outlier detection for Multidimensional Time Series using Deep Neural Networks," 2018 19th IEEE International Conference on Mobile Data Management (MDM), 2018.

[13] J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, "USAD," Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020.

[14] C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen, and N. V. Chawla, "A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, pp. 1409–1416, 2019.

[15] M. Munir, S. A. Siddiqui, A. Dengel, and S. Ahmed, "Deepant: A deep learning approach for unsupervised anomaly detection in time series," IEEE Access, vol. 7, pp. 1991–2005, 2019.

[16] H. Zhao, Y. Wang, J. Duan, C. Huang, D. Cao, Y. Tong, B. Xu, J. Bai, J. Tong, and Q. Zhang, "Multivariate time-series anomaly detection via graph attention network," 2020 IEEE International Conference on Data Mining (ICDM), 2020.

[17] T. Cover and P. Hart, "Nearest neighbor Pattern Classification," IEEE Transactions on Information Theory, vol. 13, no. 1, pp. 21–27, 1967.

[18] I. T. Jolliffe and J. Cadima, "Principal component analysis: A review and recent developments," Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 374, no. 2065, p. 20150202, 2016.

[19] A. Ben-Hur, "Support vector clustering," Scholarpedia, vol. 3, no. 6, p. 5187, 2008.

[20] R. J. Hyndman and Y. Khandakar, "Automatic time series forecasting: Theforecastpackage forr," Journal of Statistical Software, vol. 27, no. 3, 2008.

[21] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[22] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," International conference on learning representations, 2018.

[23] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," arXiv preprint arXiv:1710.10903, 2018.

[24] T. N. Kipf and M. Welling, "Semi-supervised classification with graph Convolutional Networks,"arXiv preprint arXiv:1609.02907, 2017.

[25] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," arXiv.org, 2014, [Online]. Available: https://arxiv.org/abs/1312.6114v10..

[26] C. dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," Proceedings of the 25th International Conference on Computational Linguistics, pp. 69–78, 2014.

[27] H. Ren, B. Xu, Y. Wang, C. Yi, C. Huang, X. Kou, T. Xing, M. Yang, J. Tong, and Q. Zhang, "Time-series Anomaly detection service at Microsoft," Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019.

[28] H. Touvron, P. Bojanowski, M. Caron, M. Cord, A. El-Nouby, E. Grave, G. Izacard, A. Joulin, G. Synnaeve, J. Verbeek, and H. Jegou, "RESMLP: Feedforward Networks for image classification with data-efficient training," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1–9, 2022.

[29] K. Kellogg, S. Thurman, W. Edelstein, M. Spencer, G.-S. Chen, M. Underwood, E. Njoku, S. Goodman, and B. Jai, "NASA's Soil Moisture Active Passive (SMAP) observatory," 2013 IEEE Aerospace Conference, 2013.

[30] A. P. Mathur and N. O. Tippenhauer, "Swat: A water treatment testbed for research and training on ICS Security," 2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater), 2016.

[31] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of IR techniques," ACM Transactions on Information Systems, vol. 20, no. 4, pp. 422–446, 2002.

[32] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line," Proceedings of the 24th International Conference on World Wide Web, 2015.

[33] D. Wang, P. Cui, and W. Zhu, "Structural Deep Network embedding," Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.

[34] J. Wang, P. Huang, H. Zhao, Z. Zhang, B. Zhao, and D. L. Lee, "Billion-scale commodity embedding for e-commerce recommendation in Alibaba," Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018.

Kenan Qin received her B.Sc degree in Computer Science from Shaanxi Normal University, Xian,China, in 2020. She is currently pursuing the Master degree in Computer Science with Shaanxi Normal University, Xian, China. Her research interests include anomaly detection in complex network systems.

Mengfan Xu received his PhD degree in computer system and Structure from Xidian University. His main research interest is to combine machine learning, cryptography and other methods to study network security and data security in different service scenarios such as Internet of vehicles, industrial Internet of Things and federal learning. Relevant representative results have been collected by IEEE TVT, IEEE TDSC, Information Science and other journals.

Bello Ahmad Muhammad received his B.Sc. and M.Sc. degrees in computer science from Bayero University, Kano, Nigeria, in 2010 and 2015, respectively. He is currently pursuing the Ph.D. degree in Computer Science with Shaanxi Normal University, China. He is also working with the University Library, Bayero University, Kano, Nigeria. His research interests include Learning style detection, and graph representation learning.

Jing Han received her B.Sc degree in Computer Science from Shaanxi Normal University, Xian,China, in 2021. She is currently pursuing the Master degree in Computer Science with Shaanxi Normal University, Xian, China. Her research interests include representation learning and anomaly detection of complex networks.