2022 IEEE International Conference on Prognostics and Health Management (ICPHM) | 978-1-6654-6615-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICPHM53196.2022.9815702

A Multivariate Time Series Anomaly Detection Method Based on Generative Model

1st Shaowei Chen
School of Electronic Information
Northwestern Polytechnical University
Xi 'an, P.R.China
cgong@nwpu.edu.cn

2nd Fangda Xu
AVIC Xi'an Flight Automatic Control Research Institute
Xi 'an, P.R.China
xudada@mail.nwpu.edu.cn

3rd Pengfei Wen
School of Electronic Information
Northwestern Polytechnical University
Xi 'an, P.R.China
wenpengfei@mail.nwpu.edu.cn

4th Shuaiwen Feng
School of Electronic Information
Northwestern Polytechnical University
Xi 'an, P.R.China
869404723@qq.com

5th Shuai Zhao
Department of Energy Technology
Aalborg University
Aalborg 9220, Denmark
szh@et.aau.dk

***Abstract*—Modern equipment is complex in structure, large in scale and highly integrated, in order to solve the problems of high dimension and a large amount of data collected by equipment, a multivariate time series anomaly detection method based on the deep generative model (DG-MTAD) has been proposed in this paper. Long short-term memory (LSTM) network is used to optimize the model structures of autoencoder (AE) and generative adversarial networks (GAN). While extracting time information, the bidirectional mapping of data between multi-dimensional feature space and low-dimensional latent space is completed. The reconstruction of normal time series in latent space is realized by GAN, and then the combination of generating loss and discriminant loss is calculated as the anomaly score, so as to realize the anomaly detection of multivariate time series. Experiments on the high-dimensional engine degradation monitoring data set published by NASA show that the accuracy of the method is over 90%.**

Keywords—anomaly detection, time series analysis, GAN, AE, LSTM

I. Introduction

In recent years, with the accelerated penetration of the digital revolution into various fields, the digital transformation of equipment has been ushered. In the process of data collection and status monitoring of equipment information such as vibration, current, voltage, temperature and stress by different kinds of sensors, a large number of multivariate time series data are produced [1-3]. At the same time, with the development of a new generation of information technologies such as the Internet of Things, artificial intelligence, and computers, the dimension and capacity of device data will further increase [4]. In order to more comprehensively monitor the working status of the equipment, ensure the reliability and safety of the equipment during operation, and reduce the economic losses caused by equipment failures, it is necessary to carry out accurate and timely research on multivariate time series anomaly detection for equipment [5].

This work was supported by the Laboratory of Science and Technology on Integrated Logistics Support, National University of Defense Technology.

However, in the past, the equipment anomaly detection technology was mostly aimed at univariate time series anomaly detection, only considering the impact of a specific indicator on the system health status, and lacked comprehensive judgment on multiple indicators, resulting in false positives in the detection results, which affects the accuracy of detection results [6-8]. In this regard, this paper proposes an anomaly detection method for multivariate time series based on a deep generative model (DG-MTAD). This algorithm uses encoder and Generative Adversarial Networks (GAN) to realize bidirectional mapping of data between multidimensional feature space and low dimensional latent space. Additionally, it adds two discriminators to improve the training efficiency of the model. The Long Short Term Memory (LSTM) network is used to optimize the model structure to learn the time series features. Finally, the generating loss and discriminant loss of the GAN model are combined as the anomaly score to realize anomaly judgment. This paper introduces the detailed working principle and process of the proposed algorithm, and uses the aircraft engine case published by NASA for experimental verification that proves the accuracy and timeliness of our method.

II. Related work

A. Multivariate time series

This paper focuses on anomaly detection of multivariate time series, in which each sample is a group of time series, and the length of each group of time series may not be the same. Formula (1) can be used to define a group of univariate time series:

$$T = t_1, t_2, \cdots, t_n \tag{1}$$

Where n is the length of the univariate time series. Correspondingly, the m-dimensional multivariate time series can be written as the following formula:

$$\begin{aligned} T_1 &= t_{11}, t_{12}, \cdots, t_{1n_1} \\ T_2 &= t_{21}, t_{22}, \cdots, t_{2n_2} \\ &\cdots \\ T_m &= t_{m1}, t_{m2}, \cdots, t_{mn_m} \end{aligned} \tag{2}$$

In machine learning, dimension refers to the number of features in a data set. When the number of features in a dataset is too large, it will impact the effectiveness of model training, resulting in dimension disaster [9]. Moreover, abnormal data often show more obvious abnormal features in low-dimensional space, while in high-dimensional space, abnormal features are relatively hidden and not easy to be found, which brings great challenges to accurate and timely equipment anomaly detection [10].

In order to solve the problem of high dimensional complexity in multivariate time series anomaly detection, the data need to be reduced in dimension first. The existing dimensionality reduction methods are mainly divided into two types: feature selection and feature extraction. However, traditional dimensionality reduction methods, such as Principal Component Analysis (PCA), Hidden Markov Model (HMM), etc., have high requirements on data and cannot capture the nonlinear relationship between variables well [11][12]. With the development of neural networks, many deep generative models have also been used to deal with multivariate time series feature engineering. Compared with traditional methods, deep generative algorithms have a strong ability to automatically extract efficient representations from complex features, especially in dealing with nonlinear and high-dimensional data. In this paper, we use autoencoder (AE) and GAN, aiming to solve the problem of device anomaly detection in multivariate time series.

B. LSTM

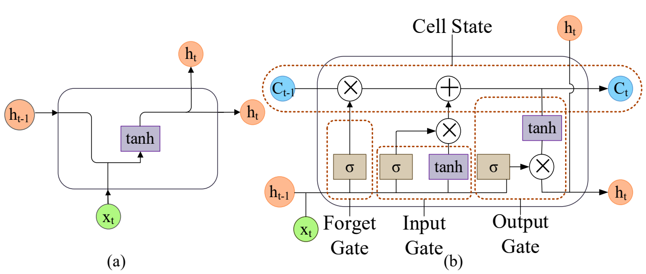

Fig. 1. Comparison of the internal structure of RNN (a) and LSTM (b) cells

Recurrent Neural Network (RNN) is used for modeling sequential data. As a special kind of RNN, LSTM has the ability to maintain both long-term memory and short-term memory of parameter information. As shown in Fig. 1, different from RNN, LSTM adds gate structure and unit state line to avoid gradient disappearance and gradient explode during training [13]. The detailed calculation formulas related to the gate structure and unit state output in LSTM are as follows:

Forgotten Gate: $$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{3}$$

Input gate: $$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \tag{4}$$

Output gate: $$o_t = \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right) \tag{5}$$

Unit status update1: $$\tilde{C}_t = \tanh\left(W_c \cdot [h_{t-1}, x_t] + b_c\right) \tag{6–1}$$

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \tag{6–2}$$

LSTM cell output: $$h_t = o_t \times \tanh(C_t) \tag{7}$$

Where x_t is the input, h_t is the output, and C_t is the output of the cell state in time t. W and b represent weights and biases respectively.

C. AE

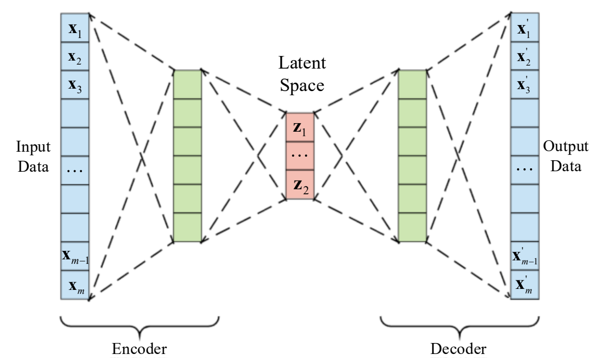

Fig. 2. A typical autoencoder network structure

An autoencoder is an artificial neural network that learns from end-to-end representation learning. By learning to change the weight of each neuron to reduce the difference between input and reconstructed output, an efficient representation of input data can be obtained in hidden space. It consists of an encoder network and a decoder network. The process of obtaining an efficient representation from the input data is called encoding, and the process of reconstructing the input data from the efficient representation is called decoding [14]. Here, we refer to the sample space where the efficient representation of the data resides as the latent space. In order to reduce a typical autoencoder network structure are usually symmetric, as shown in Fig. 2.

The whole process of encoding and decoding can be expressed as:

Encoder: $\mathbf{z} = f(\mathbf{x}), \quad \dim(\mathbf{z}) = d \cap \dim(\mathbf{x}) = m \cap d < m$ (8)

Decoder: $\mathbf{x}' = g(\mathbf{z}), \quad \dim(\mathbf{z}) = d \cap \dim(\mathbf{x}') = m \cap d < m$ (9)

Objective function: $f, g = \arg\min_{f,g} \|\mathbf{x} - \mathbf{x}'\|^2$ (10)

Among them, f and g represent the mapping function from the encoder to the latent space and the mapping function from the latent space to the decoder, respectively. The entire autoencoder training process is shown in Fig. 3.

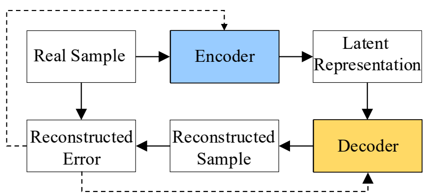

Fig. 3. The training process of AE

D. GAN

GAN is a powerful high-dimensional data modeling framework, which is often used to model complex and high-dimensional data distribution. It has achieved great success in the fields of image recognition, image generation, image segmentation, style transfer and so on, and has been gradually promoted to the application of anomaly detection in timing signals and images [15]. It consists of a generator (G) and a discriminator (D). The essential idea is to learn the feature distribution through the dynamic game [16].

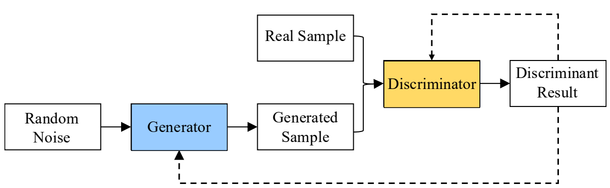

Fig. 4. A typical training process of GAN

A typical GAN training process is shown in Fig. 4. The main task of the generator is to learn the distribution of real sample data, generate data with a distribution that is as similar as possible to the real sample data, and continuously optimize according to the judgment results of the discriminator, and finally achieve the effect of deceiving the discriminator. However, the discriminator's task is to judge the probability that the input data comes from the real sample data rather than the generator, and optimize its model parameters according to the discriminant results to improve its discriminant ability. The objective function of the GAN model is as follows:

$$\min_G \max_D V(D,G) = E_{x\sim Pdata(x)}[\log(D(x))] + E_{z\sim PZ(z)}[\log(1-D(G(z)))] \quad (11)$$

$$\min_G V(D,G) = E_{z\sim PZ(z)}[\log(1-D(G(z)))] \quad (12)$$

$$\max_D V(D,G) = E_{x\sim Pdata(x)}[\log(D(x)] + E_{z\sim PZ(z)}[\log(1-D(G(z)))] \quad (13)$$

Among them, input sample data $x \sim Pdata(x)$ sampled from the real training set, reconstructed data $x' \sim PG(x)$, random noise $z \sim PZ(z)$, and the model is constrained by the minimum-maximization adversarial loss $V(D,G)$.

The vanilla GAN generator and discriminator were composed of the full connection layer. Deep convolutional generative adversarial networks (DCGAN) introduces the convolution operation into GAN to further improve the learning ability of the model [17]. Bidirectional generative adversarial networks (BiGAN) can not only generate high-dimensional real data from low-dimensional distribution through generators, but also learn the inverse mapping by encoder [18]. The proposed method is inspired by BiGAN and further improves GAN networks.

III. Proposed Approach

In this paper, we propose a multivariate time series anomaly detection based on deep generative model method, named DG-MTAD. The DG-MTAD algorithm proposes a combination of two powerful deep generative models—AE and GAN to solve the problem of high-dimensional complexity of multivariate data. The encoder is embedded in the framework of adversarial training to realize the reverse learning from the sample data space to the hidden space, avoiding the model loss in accuracy and real-time. And two additional discriminators are added to the training process to improve the model stability. At the same time, the LSTM network is used as the basic structure of the encoder, generator and discriminator models in the algorithm, so as to better solve the problem of time series feature extraction of time series data. In addition, the DG-MTAD algorithm adopts the training method of semi-supervised learning. During the training process, only normal data is used, so that this algorithm can also perform well in scenarios with unbalanced datasets or lack of data labels.

The overall framework of the DG-MTAD algorithm is shown in Fig. 5. It consists of an encoder (E), a generator (G) and three discriminators (D1, D2, D3). There are two stages in

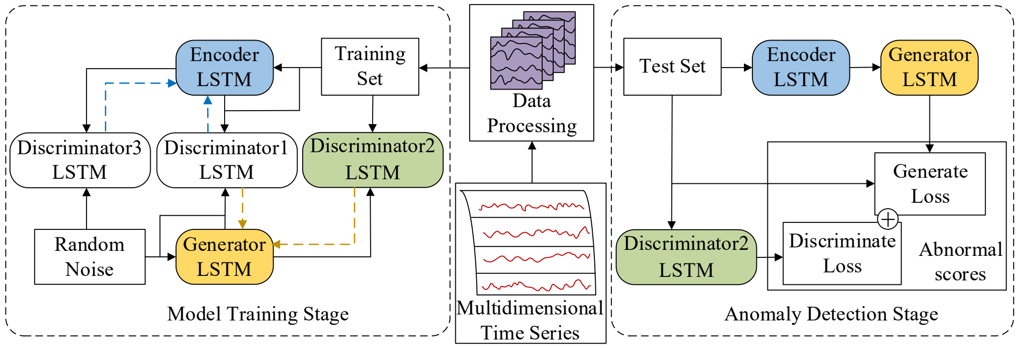

Fig. 5. The framework of DG-MTAD

this algorithm: the model training stage and the anomaly detection stage.

A. The Stage of Model Training

In the model training stage, firstly, the processed normal time series data is input into the encoder, and the randomly sampled noise data in the latent space is input into the generator simultaneously. The generator can map the random noise $\mathbf{Z}$ in the latent space to the generated sample $G(\mathbf{Z})$ in the sample data space. At the same time, the encoder can encode the real sample into an efficient representation in the latent space.

There are three discriminators used to constrain the output of the encoder and generator. The discriminator 1 (D1) is used to evaluate the possibility that the sample comes from the real data $\mathbf{X}$ instead of the generator $G(\mathbf{Z})$. Combine $\mathbf{X}$ with $E(\mathbf{X})$ in the sample data space and combine $\mathbf{Z}$ with $G(\mathbf{Z})$ in the hidden space as input in D1 training. The adversarial training is carried out through the following objective function:

$$\min_{E,G}\max_{D_1} V(D_1,E,G)=E_{x\sim Pdata(x)}[\log(D_1(x,E(x)))] + E_{z\sim PZ(z)}[\log(1-D_1(G(z),z))] \tag{14}$$

The discriminator 2 (D2) is used to evaluate the similarity between $G(E(\mathbf{X}))$ and the real data $\mathbf{X}$. Discriminator 3 (D3) can make $E(G(\mathbf{Z}))$ and $\mathbf{Z}$ as similar as possible. the objective functions of D2 and D3 during training are as follows:

$$\min_{E,G}\max_{D_2} V(D_2,E,G)=E_{x\sim Pdata(x)}[\log(D_2(x,x))] + E_{z\sim PZ(z)}[\log(1-D_2(x,G(E(x))))] \tag{15}$$

$$\min_{E,G}\max_{D_3} V(D_3,E,G)=E_{x\sim Pdata(x)}[\log(D_3(z,z))]+ E_{z\sim PZ(z)}[\log(1-D_3(z,E(G(x))))] \tag{16}$$

Therefore, the overall objective function of the DG-MTAD algorithm includes the loss functions of the above three discriminators, which are specifically defined as follows:

$$\min_{E,G}\max_{D_1,D_2,D_3} V(D_1,D_2,D_3,E,G)=V(D_1,E,G)+V(D_2,E,G) +V(D_3,E,G) \tag{17}$$

B. The Stage of Anomaly detection

In the stage of anomaly detection, the test set including normal and abnormal data are input into the trained model. The trained E, G and D2 are used for anomaly detection. Using the backward learning of the E, the test data are mapped from the sample data space to the hidden space, and then input into the GAN. According to the generative loss and discrimination loss, the corresponding anomaly score is calculated, so as to realize the anomaly determination of multivariable time series. By comparing the difference between the generated sample and the real sample in GAN model, the reconstruction error of normal sample is small while that of abnormal sample is large. The abnormal score calculation of the DG-MTAD algorithm consists of the following two parts:

- Discrimination loss: Since the trained D2 has high sensitivity, it can realize the distinction between real time series and generated data, so it can be directly used as a tool for anomaly detection, and the discrimination loss in the anomaly score is expressed by the following formula:

$$L_{dis}=D_2(\mathbf{X}_{test}) \tag{18}$$

- Generate loss: The trained encoder can map the real samples to the latent space, use the trained generator to reconstruct the latent space representation of the samples. Then compare the reconstructed samples with the real samples to obtain the generation loss of the model. The calculation formula of the generation loss is as follows:

$$L_{ger}=\sum\left|\mathbf{X}-G(E(\mathbf{X}))\right| \tag{19}$$

Therefore, the anomaly score calculation formula of the DG-MTAD algorithm is as follows:

$$L = \alpha L_{dis} + L_{ger} \tag{20}$$

Among them, α is the weight parameter of the discriminant loss, which is used to adjust the proportion of the influence of the discriminative loss and the generation loss on the abnormal score. The calculated anomaly score is compared with the threshold θ to complete the anomaly determination.

Due to the need for sliding window processing in sample preprocessing, samples at some moments participate in multiple abnormal score calculation processes, and sample abnormal scores at some moments are only calculated once. We use the average abnormal score of samples at each moment as the abnormal score. The pseudo code of DG-MTAD algorithm is as follows:

TABLE I. DG-MTAD ALGORITHM PROCESS

Algorithm 1: DG-MTAD Multivariate Time Series Anomaly Detection

Inputs: $\mathbf{X}_{train}$, $\mathbf{Z}$, $\mathbf{X}_{test}$

Outputs: The anomaly score L

Model training:

1. Initialize the model parameters of E, G, D1, D2, D3
2. Input $\mathbf{X}_{train}$ into the E as well as $\mathbf{Z}$ into the G
3. Adversarial training is performed according to formula (17):

   $E(\mathbf{X}_{train}) \to \mathbf{Z}$, $G(\mathbf{Z}) \to \mathbf{X}$, $E(G(\mathbf{Z})) \to \mathbf{Z}$, $G(E(\mathbf{X})) \to \mathbf{X}$
4. Model parameter update

Abnormal detection:

1. Obtain trained model parameters of E, G, D2 and the threshold θ
2. Calculate L_{dis} according to formula (19)
3. Calculate L_{ger} according to formula (20)
4. Calculate anomaly scores according to $L = \alpha L_{dis} + L_{ger}$
5. Calculate the average anomaly score for each time series point
6. **for** i in $len(\mathbf{X}_{test})$:

   if $L_i > \theta$:

   $\mathbf{x}_i$ abnormal

   else:

   $\mathbf{x}_i$ normal
7. **end**

IV. EXPERIMENTAL ANALYSIS

A. Dataset Description and Processing

The data set selected in this experiment is the aircraft gas turbine engine data set publicly released by NASA. This dataset is derived from the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS), using a wealth of sensors to record the health of multiple complex engine equipment full life cycle monitoring data from operation to system failure [19]. The dataset is divided into four subsets - FD001, FD002, FD003, and FD004, based on engine operating conditions and the number of failure modes. We use FD001 as the dataset used in our experiments. The dataset consists of 100 aircraft gas turbine engines and records 21 sensor changes from the start of operation to equipment failure for each aircraft gas turbine engine. Taking the No. 14 engine as an example, Fig. 6 shows the variation trend of the variable information collected by the 21 sensors of the No. 14 engine with the monitoring time (unit is the monitoring time). Different colors represent different sensors.

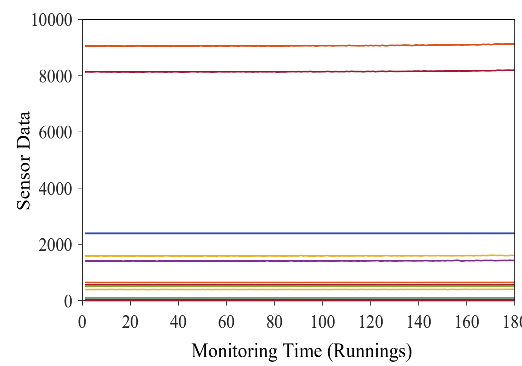

Fig. 6. No. 14 engine data observation

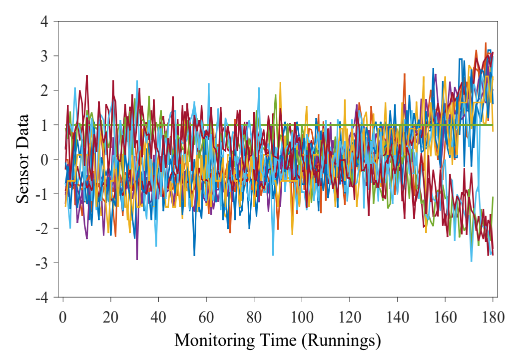

Fig. 7. No. 14 engine data after the data normalization

- Data normalization: There is a large difference of order of magnitude between different variables in the original data set, so it is difficult to observe the change information of sensor features over time and cannot be directly compared horizontally. Therefore, data need to be normalized first. We use zero-mean normalization for the sequence $x_1, x_2, \cdots, x_n$, the formula is as follows:

$$y_i = \frac{x_i - \overline{x}}{s} \tag{21}$$

Here, $\overline{x}$ is the mean of data in the training set, s is the standard deviation of data in the training set. So a new standardized sequence data $y_1, y_2, \cdots, y_n$ is obtained. When the sensor data is constant, we use the following formula for normalization:

$$y_i = \frac{x_i}{\max(x_i)} \tag{22}$$

The results after the data normalization is shown in Fig. 7.

- Data label calibration: The engine operation process is divided into two stages: normal stage and abnormal stage. For each engine, the sampling data of the first 50% of the specified operation cycle are trained as normal data.
- Data set division: Among the 100 engines, we randomly select 20 engines, and take the first 50% of the data collected by the sensors in each engine data as the training set, and the data in the training set are normal. Then 30 engines were randomly selected from the remaining engines as the validation set, which was used to determine the anomaly score threshold of θ. Finally, the remaining 50 engines are used as the test set data, which contains both normal and abnormal data.
- Data slice: The sliding window method is used to slice the data set. According to experience, for the FD001 engine data set used, $k = 25$ is used as the window width parameter of the sliding window in the experiment, and $s = 1$ is used as the moving step parameter in the sliding window method, and the processed subsequence is obtained.

B. Experimental Setup

In order to process time series, generator, discriminator and encoder models in LSTM network construction algorithm are proposed. The network model layers of each structure are as follows:

TABLE II. DG-MTAD MODEL STRUCTURE

Model	Network Type	Number of network layers	Number of hidden layers
Encoder	LSTM	3	128+64+32
Builder	LSTM	3	32+64+128
Discriminator 1	LSTM	1	100
Discriminator 2	LSTM	1	100
Discriminator 3	LSTM	1	100

Table 3 lists the determined model parameters in the DG-MTAD algorithm experiment, and the remaining unspecified model parameters are default values.

TABLE III. OTHER PARAMETER SETTINGS

Model parameters	Setting
Batch size	128
Learning rate	0.0001
The maximum number of iterations	300
Optimizer	Adam

In addition, the DG-MTAD proposed in this paper combines the GAN, AE and LSTM models. In order to prove the effectiveness of the method, it is compared with the current algorithms that are widely used and have similar model structures. Among them, the LSTM-VAE algorithm uses LSTM to replace the VAE. The feedforward network is an improved model combining LSTM and VAE; the MAD-GAN algorithm uses LSTM as the structure of the generator and discriminator in the GAN network to solve the interaction problem between multiple variables of time series data; in the LSTM-VAE-GAN algorithm Similarly, LSTM, VAE and GAN are combined to jointly train the encoder, generator and discriminator, while utilizing the mapping ability of the encoder and the discriminant ability of the discriminator. In addition, the model parameter settings used for control in the experiment are the same as the DG-MTAD method to achieve the purpose of controlling variables.

C. Experimental results and analysis

During the experiment, the accuracy and timeliness of the model performance were first verified. In order to better verify the accuracy and timeliness of the proposed DG-MTAD algorithm, the more mature popular algorithms of LSTM-VAE, MAD-GAN and VAE-GAN are used as comparison algorithms. We trained the DG-MTAD algorithm and the above-mentioned comparison model algorithm for many times, and counted various indicators in the nearly 10 model training process. In the model accuracy evaluation process, the precision, recall and F1 score are used as the evaluation indicators of the model. When verifying the timeliness of the model, the average time spent in each training round (unit: seconds) is used as the evaluation indicator. Validated on the processed engine data set, the validation results are as follows:

TABLE IV. ACCURACY COMPARISON

Evaluation indicators	DG-MTAD	LSTM-VAE	MAD-GAN	VAE-GAN
Accuracy	0.922	0.746	0.879	0.930
Recall	0.934	0.852	0.830	0.892
F1 Score	0.923	0.800	0.887	0.910

By analyzing the comparison results of the accuracy and timeliness of the DG-MTAD algorithm, the following conclusions can be drawn:

- The order of model accuracy performance is DG-MTAD> LSTM-VAE-GAN> MAD-GAN> LSTM-VAE. In addition, the DG-MTAD model proposed in this paper reaches more than 90% in the three evaluation indexes of accuracy, recall rate and F1 score, which proves the effectiveness of the algorithm for multivariable time series anomaly detection

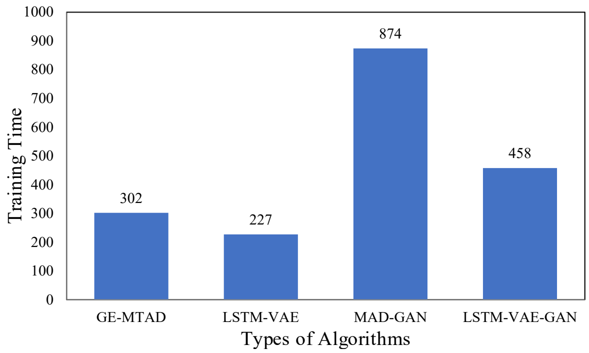

Fig. 8. Time spent on training

- The order of model training time spent is MAD-GAN> LSTM-VAE-GAN>DG-MTAD> LSTM-VAE. It is concluded that the LSTM-VAE model is the simplest, so the training time is the shortest; the MAD-GAN model has the most complex algorithm route and the longest training time; and the LSTM-VAE-GAN model is similar in structure to the DG-MTAD model, and both use encoder reverse However, since the proposed DG-MTAD method adds two additional discriminators, the encoder and generator are more purposeful in the training process, which shortens the training time.

- Combining the accuracy comparison structure in Table 4 and the timeliness comparison results in Fig. 8, and evaluating the above four methods, it is concluded that the proposed DG-MTAD method can achieve engine multivariate time series anomaly detection on the basis of good timeliness. The best performance in the accuracy problem, the experiments demonstrate the superiority of the DG-MTAD method.

The DG-MTAD method proposed in this paper is a superposition model of multi-model fusion. In order to further prove the advantages of model stacking and discuss the influence of different stacking structures on model performance, the proposed DG-MTAD algorithm was subjected to ablation experiments. In the experiment, the GAN model was used as the baseline model (Baseline) for model stacking. Precision, recall and F1 score are used to evaluate model performance, and each model is run 10 times repeatedly. The mean values of evaluation indicators in the experimental results are as follows:

TABLE V. DG-MTAD MODEL STRUCTURE

Evaluation indicators	Baseline	LSTM + Baseline	AE + Baseline
Accuracy	0.704	0.881	0.844
Recall	0.939	0.850	0.810
F1 Score	0.805	0.865	0.827

- The experimental results in Table 5 Baseline and LSTM + Baseline, AE + Baseline and LSTM + AE + Baseline are grouped and compared, which proves that adding LSTM network can greatly improve the performance of the model.

- Change the combination method and compare Baseline with AE + Baseline, LSTM + Baseline with LSTM + AE + Baseline, and the results show that the performance of the encoder model has been improved to a certain extent.

- The experimental results demonstrate the effectiveness of the model stacking. The F1 of the DG-MTAD obtained after superposition reaches 0.923, which is much higher than the Baseline model, and the effect is obvious. It shows that the proposed DG-MTAD algorithm combines the advantages of LSTM in processing time series time series features, the advantages of encoder dimensionality reduction and noise reduction, and the advantages of GAN The adversarial training in the model obtains a strong data distribution fitting ability and high-dimensional data learning ability, thereby achieving a better detection effect on multivariate time series anomaly detection problems

V. CONCLUSIONS

In this paper, a deep generative multivariate time series anomaly detection method, DG-MTAD, which can solve high-dimensional time series anomaly detection, is established. The algorithm consists of a generator, an encoder, and 3 discriminators. It performs joint adversarial training in a semi-supervised learning manner. The algorithm combines the advantages of LSTM in processing time series features, the advantages of encoder dimensionality reduction and noise reduction, and the GAN model. The medium-level adversarial training obtains strong data distribution fitting and high-dimensional data learning advantages, and finally uses the discriminative loss and the generation loss to calculate the average anomaly score at each moment to realize the anomaly discrimination of multivariate time series. Experiments are carried out on the multivariate engine data set released by NASA. Through the comparison experiment with other popular methods and the ablation experiment of DG-MTAD itself, it is proved that the proposed DG-MTAD can take into account the accuracy and timeliness at the same time. The multivariate time series anomaly detection problem has better detection results.

REFERENCES

[1] Liu, Zepeng, and Long Zhang. "A review of failure modes, condition monitoring and fault diagnosis methods for large-scale wind turbine bearings." Measurement 149 (2020): 107002.

[2] Gangsar, Purushottam, and Rajiv Tiwari. "Signal based condition monitoring techniques for fault detection and diagnosis of induction motors: A state-of-the-art review." Mechanical systems and signal processing 144 (2020): 106908.

[3] Gundewar, Swapnil K., and Prasad V. Kane. "Condition monitoring and fault diagnosis of induction motor." Journal of Vibration Engineering & Technologies 9.4 (2021): 643-674.

[4] Sahal, Radhya, John G. Breslin, and Muhammad Intizar Ali. "Big data and stream processing platforms for Industry 4.0 requirements mapping

for a predictive maintenance use case." Journal of manufacturing systems 54 (2020): 138-151.

[5] Zhang, Chuxu, et al. "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. 2019.

[6] Braei, Mohammad, and Sebastian Wagner. "Anomaly detection in univariate time-series: A survey on the state-of-the-art." arXiv preprint arXiv:2004.00433 (2020).

[7] Kao, Jian-Bin, and Jehn-Ruey Jiang. "Anomaly detection for univariate time series with statistics and deep learning." 2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE). IEEE, 2019.

[8] Geiger, Alexander, et al. "TadGAN: Time series anomaly detection using generative adversarial networks." 2020 IEEE International Conference on Big Data (Big Data). IEEE, 2020.

[9] Wang, Yongqing, and Qingxiu Li. "Review on the studies and advances of machine learning approaches." TELKOMNIKA Indonesian Journal of Electrical Engineering 12.2 (2014): 1487-1494.

[10] Zheng, Jian, et al. "An irrelevant attributes resistant approach to anomaly detection in high-dimensional space using a deep hypersphere structure." Applied Soft Computing 116 (2022): 108301.

[11] Huang, Tingshan, Harish Sethu, and Nagarajan Kandasamy. "A new approach to dimensionality reduction for anomaly detection in data traffic." IEEE Transactions on Network and Service Management 13.3 (2016): 651-665.

[12] Poh, Soon-Chang, et al. "LSTM and HMM comparison for home activity anomaly detection." 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). IEEE, 2019.

[13] Sherstinsky, Alex. "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network." Physica D: Nonlinear Phenomena 404 (2020): 132306.

[14] Chen, Zhaomin, et al. "Autoencoder-based network anomaly detection." 2018 Wireless Telecommunications Symposium (WTS). IEEE, 2018.

[15] Zhao, Zhengli, et al. "Image augmentations for gan training." arXiv preprint arXiv:2006.02595 (2020).

[16] Zenati, Houssam, et al. "Efficient gan-based anomaly detection." arXiv preprint arXiv:1802.06222 (2018).

[17] Gao, Fei, et al. "A deep convolutional generative adversarial networks (DCGANs)-based semi-supervised method for object recognition in synthetic aperture radar (SAR) images." Remote Sensing 10.6 (2018): 846.

[18] Zhang, Zhirui, et al. "Bidirectional generative adversarial networks for neural machine translation." Proceedings of the 22nd conference on computational natural language learning. 2018.

[19] Saxena, Abhinav, et al. "Damage propagation modeling for aircraft engine run-to-failure simulation." 2008 international conference on prognostics and health management. IEEE, 2008.