Telemetry Data-Based Spacecraft Anomaly Detection With Spatial–Temporal Generative Adversarial Networks

Jinsong Yu, *Member, IEEE*, Yue Song, Diyin Tang, *Member, IEEE*, Danyang Han, and Jing Dai

***Abstract*— The telemetry data obtained from an on-orbit spacecraft contain important information to indicate anomaly of the spacecraft. However, the large number of monitoring variables and the large amount of data points, as well as the lack of prior knowledge about anomaly due to complicated structure of spacecraft and its working conditions, pose great challenge to the anomaly detection. This article proposes an anomaly detection algorithm based on a spatial–temporal generative adversarial network (GAN) for the anomaly detection in telemetry data. This algorithm establishes a GAN-based model combining convolutional neural network (CNN) and long short-term memory (LSTM) to extract spatial and temporal features of the telemetry data, which facilitates the automatic and simultaneous representation of nonnegligible time-related characteristics of a monitoring variable and complex correlation between variables. Using these features, many kinds of anomalies including multivariate anomalies and contextual anomalies can be detected. Moreover, an anomaly score specifically designed to fit the GAN-based algorithm is also proposed to evaluate the possibility of anomaly by weighted fusion of the generator metric and the discriminator metric, which is proved to be significantly helpful to the accuracy of anomaly detection. Finally, experiments on one real telemetry data set and two public telemetry data sets are conducted, by which the proposed anomaly algorithm is demonstrated to be effective and accurate in detecting outliers in telemetry data. Comparison with three other state-of-the-art methods also reveals the advantages of our proposed algorithm.**

***Index Terms*— Anomaly detection, anomaly score, generative adversarial networks (GANs), multivariate time series, telemetry data.**

I. Introduction

THIS article is an extension of the previously published proceedings paper [1]. Anomaly detection from telemetry data which are sent continuously to the ground is an applicable way to discover early the failures, either observable or potential, in an on-orbit spacecraft [2]. However, the telemetry data have many monitoring variables (usually a couple of thousands of variables), and each of them can be considered as a time series with a quite high sampling frequency. Thus, the amount of data is too large for rule-based anomaly detection systems to be very effective. Although some good results based on rule-based methods were obtained [3], [4], with the increasing complexity of spacecraft and its working conditions, the rule-setting becomes more and more difficult, mainly due to the lack of prior knowledge and the timely update to new situations.

Recently, data mining methods based on machine learning and deep learning have been widely used in anomaly detection of telemetry data [5], [6], which do not require prior knowledge determined by experts and meanwhile are capable of automatically adapting to new situations. NASA Jet Propulsion Laboratory realized the anomaly detection of telemetry data with long short-term memory (LSTM) and proposed a nonparametric anomaly threshold setting method [7]. However, this method is only for one variable in telemetry data, which means detecting anomalies in multiple variables requires a training detection algorithm for each variable and invoking multiple train models simultaneously in online detection, leading to inefficient detection process and possible failure of meeting real-time requirements. In addition, this method cannot identify the multivariate anomaly caused by the change of the correlation between multiple variables, i.e., failures such as one failure resulting in multiple anomalies cannot be identified. Therefore, methods considering multiple variables and their correlations simultaneously were proposed. For example, NASA Ames Research Center proposed an inductive monitoring system (IMS) based on the idea of clustering the behavior of normal data and anomalies [8]. Weng and Liu [9] used isolation forest (Iforest) to divide binary tree step by step until all the data reach the leaf node, where the leaf node of an outlier is close to the root node. However, they do not take advantage of temporal features in telemetry data, which indicates spacecraft contextual anomalies such as slow degradations with operating time are difficult to discover. Therefore, considering both spatial and temporal features in telemetry data is of great significance to the accuracy and efficiency of spacecraft anomaly detection.

Among all anomaly detection algorithms, some deep generative models capable of mining multivariable features have

Manuscript received January 17, 2021; revised March 18, 2021; accepted April 5, 2021. Date of publication April 15, 2021; date of current version May 5, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1403300 and in part by the National Natural Science Foundation of China under Grant 51875018. The Associate Editor coordinating the review process was Zhigang Liu. *(Corresponding author: Diyin Tang.)*

Jinsong Yu, Yue Song, and Diyin Tang are with the School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China (e-mail: yujs@buaa.edu.cn; songyue@buaa.edu.cn; tangdiyin@buaa.edu.cn).

Danyang Han is with the School of Instrumentation Science and Opto-electronics Engineering, Beihang University, Beijing 100191, China (e-mail: hdy_daniel@buaa.edu.cn).

Jing Dai is with the China Academy of Launch Vehicle Technology Research and Development Center, Beijing 100076, China (e-mail: buaadaij@126.com).

Digital Object Identifier 10.1109/TIM.2021.3073442

been proposed recently. For example, Su *et al.* [10] proposed an anomaly detection approach based on a variational autoencoder with a gated recurrent unit (GRU-VAE), which performed well in public telemetry data sets of NASA. In this article, we also take a deep generative model, the generative adversarial network (GAN), as the basis of our proposed anomaly detection algorithm. Combined with the generator and the discriminator, GAN extracts the data distribution of normal samples, and then anomalies can be discovered by comparing the difference between data samples and the constructed distribution of normal data [11]. A few GAN-based anomaly detection algorithms can be found in the literature. For instance, Li *et al.* [12] proposed a GAN-based multivariate anomaly detection method with an LSTM structure, which is used in cyberattacks data and water treatment data. However, those methods did not quite consider the correlations between variables and their anomaly score is not specifically designed to fit telemetry data-based anomaly detection algorithm, and thus they might not be effective enough for anomaly detection using telemetry data.

This article proposes an anomaly detection algorithm referred to as ST-GAN to detect anomalies in telemetry data. There are three main contributions of the proposed algorithm. Firstly, it constructs a special GAN-based model to explore the distribution of normal samples with a quite high accuracy such that slightly anomalous behavior can be found. Secondly, a time-scale convolutional kernel and a spatial-scale convolutional kernel are specifically designed and combined as the network structure in the GAN-based model to simultaneously extract spatial and temporal features in telemetry data, proving to be effective in identifying multivariate anomalies and contextual anomalies. Thirdly, a novel anomaly score called GDScore is proposed based on the proposed GAN-based model to determine the possibility of anomaly occurrence, which has the characteristic of automatic adjustment and will be proved effective in improving detection accuracy by our real data experiments.

The rest of this article is organized as follows. Section II presents the problem of spacecraft anomaly detection using telemetry data. Section III describes the details of the proposed ST-GAN anomaly detection algorithm. Experimental results are reported in Section IV. Section V concludes this article.

II. Problem Description

The spacecraft anomalies are generally reflected as outliers or abnormal subsequences in multiple variables in telemetry data. Examples are shown in Fig. 1. As shown in Fig. 1(a), a power failure can cause anomaly of its output voltage, current, and loads. If the simultaneous anomalies can be detected accurately, the power failure can be identified. Therefore, detecting multivariate anomaly lying in multiple variables is very important for the location of root failures. Moreover, as shown in Fig. 1(b), similar speed behavior of a bearing is defined as abnormal if it has degraded to a certain degree while defined as normal otherwise. Such contextual anomaly cannot be found by methods that do not consider temporal features.

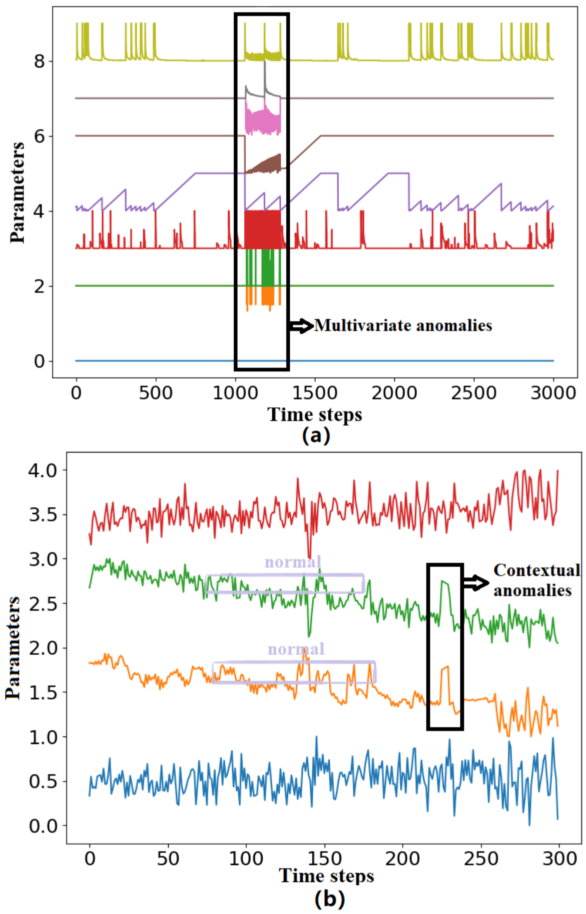

Fig. 1. Illustrative examples of anomalies in telemetry data. (a) Multivariate anomalies. (b) Contextual anomalies.

The objective of anomaly detection is to find the outliers or abnormal subsequences in a multivariate time series $\Phi \subset R^{T \times N}$, where T is the length of timestamps and N is the number of variables. Since the whole process is unlabeled, telemetry data anomaly detection is an unsupervised learning problem.

In GAN-based anomaly detection algorithm, GAN serves as the model to construct the distribution of normal data. The reconstruction deviation between the reconstructed time series and the original time series is the indicator of whether the time series is abnormal or not. Define $X_{\{T\}}$ as the original time series with a time span of T, $\hat{X}_{\{T\}}$ is the time series reconstructed by GAN-based model, then whether $X_{\{T\}}$ is abnormal or not can be obtained by

$$||X_{\{T\}} - \hat{X}_{\{T\}}|| > \tau \tag{1}$$

where $||\bullet||$ denotes the rule to calculate the deviation between $X_{\{T\}}$ and $\hat{X}_{\{T\}}$, τ is a predefined anomaly threshold. The greater the deviation between $X_{\{T\}}$ and $\hat{X}_{\{T\}}$, the higher the abnormal probability of $X_{\{T\}}$ will be.

Therefore, our proposed ST-GAN algorithm has two parts. One is a GAN-based model specifically designed to capture temporal and spatial features in normal telemetry data. In other words, the model is used to obtain $\hat{X}_{\{T\}}$ in (1). The other is the rule to calculate the deviation between $X_{\{T\}}$ and $\hat{X}_{\{T\}}$.

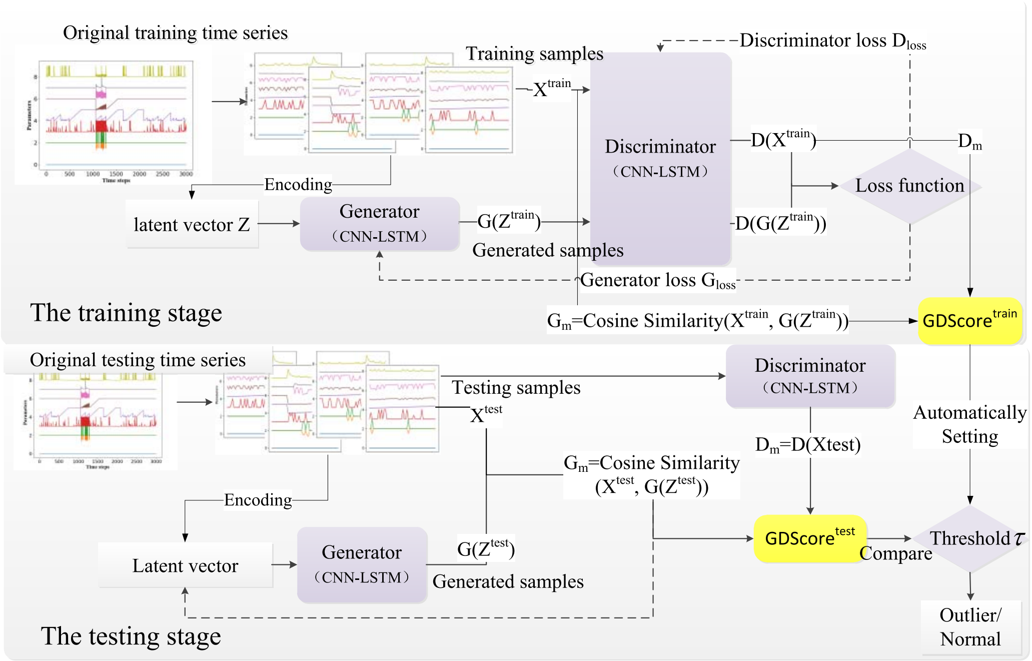

Fig. 2. Overall architecture of the proposed ST-GAN anomaly detection algorithm.

The proposed rule in our algorithm is an anomaly score called GDScore, and the minimum value of GDScore calculated from training data is determined as the anomaly threshold τ.

III. Details of the Proposed Method

In order to predict whether the observation vector $x_t \subset R^N$ of N variables of current data point t is an outlier, the historical observation subsequence $X_{t-s_w:t}$ with length s_w, which equals to $X_{\{s_w\}}$, are used to predict the anomaly state of x_t. By calculating the deviation between the reconstructed subsequence $\hat{X}_{t-s_w:t}$ and original subsequence $X_{t-s_w:t}$, the abnormal probability of x_t can be determined. Note that s_w is the length of a sliding window. The GAN-based model in our proposed detection algorithm is used to obtain the reconstructed subsequence $\hat{X}_{t-s_w:t}$, and the proposed anomaly score GDScore is used to quantify the anomaly possibility of x_t based on the deviation between $X_{t-s_w:t}$ and $\hat{X}_{t-s_w:t}$.

A. Data-Preprocessing

Firstly, split multivariate telemetry data $\Phi \subset R^{T\times N}$ of total length T and number of variables N into a training data set $\Phi_{\text{train}} \subseteq R^{T_1\times N}$, a testing data set $\Phi_{\text{test}} \subseteq R^{T_2\times N}$ and a validation data set $\Phi_{\text{valid}} \subseteq R^{T_3\times N}$, in which the validation data set is used to search the best length of sliding window. Note that all the data points in the training data set must be normal.

Next, the training data set $\Phi_{\text{train}} \subseteq R^{T_1\times N}$ is segmented into a series of subsequences $X_{\text{train}} = \{X^i_{\text{train}}, i = 1, 2, \ldots, m\} \subseteq R^{S_w\times N}$ by a sliding window, where X^i_{train} means $X_{t-s_w:t}$ when the window size is set to s_w. Given step length s_t, the number of subsequences can be calculated by $m = \lfloor (T_1 - s_w)/s_t \rfloor + 1$. Similarly, the testing data set $\Phi_{\text{test}} \subseteq R^{T_2\times N}$ is divided into subsequences $X_{\text{test}} = \{X^j_{\text{test}}, j = 1, 2, \ldots, n\}$ by a sliding window, where $n = \lfloor (T_2 - s_w)/s_t \rfloor + 1$. For verification purpose, every point in the testing data set is labeled by binary digits (1 for normal and 0 for anomalous).

B. Overall Framework of ST-GAN

The overall ST-GAN algorithm is composed of training and testing stages. Its overall architecture is presented in Fig. 2.

In the training stage, the training samples are all normal data and the GAN-based model extracts features such that spatial and temporal features of normal data can be obtained automatically. The extracted features represent the normal pattern of telemetry data. Given a set of random vectors in the latent space $Z_{\text{train}} = \{Z^i_{\text{train}}, i = 1, 2, \ldots, m\}$ encoded by the training samples X_{train} using the method in [13], the generator G implicitly defines a distribution of generated samples P_g to fit the distribution of original samples P_{data} by mapping Z_{train} to the generated samples $G(Z_{\text{train}})$. Then, $G(Z_{\text{train}})$ and X_{train} are both sent to the discriminator D for distinction. The output of the discriminator D, $D(G(Z_{\text{train}}))$ and $D(X_{\text{train}})$, are the probabilities that samples from $G(Z_{\text{train}})$ and X_{train} can be identified correctly. The network structure of the generator and the discriminator is based on CNN and LSTM, which extract temporal features and spatial features in telemetry data.

The generator and the discriminator are trained alternately with each other. The discriminator is trained to distinguish the original samples X_{train} and the generated samples $G(Z_{\text{train}})$ as much as possible, while the generator is trained to generate samples that are very similar to the original samples to mislead the discriminator. The entire optimization process can be regarded as a minimax game problem. The loss function is

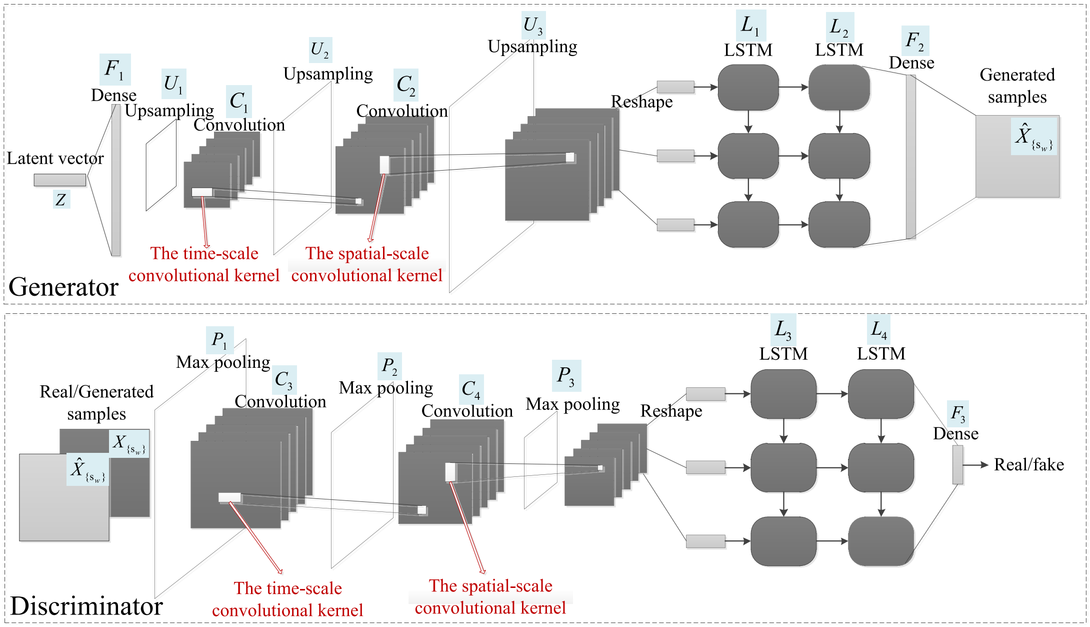

Fig. 3. Network structure of the generator and the discriminator.

defined as [11]

$$\begin{aligned}\min_G \max_D V(D, G) = {} & E_{x\sim p_{\text{data}}}[\log D(X_{\text{train}})] \\ & + E_{x\sim p_g}[\log(1 - D(G(Z_{\text{train}})))] \end{aligned} \quad (2)$$

where $x \sim p_{\text{data}}$ represents the samples from the real data distribution while $x \sim p_g$ denotes the samples from the generated data distribution. $E_{x\sim p_{\text{data}}}[\log D(X_{\text{train}})]$ represents the expectation that the original samples are judged to real by the discriminator while $E_{x\sim p_g}[\log(1 - D(G(Z_{\text{train}})))]$ denotes the expectation that the generated samples are judged to fake by the discriminator.

The performance of the discriminator and the generator is continuously improved through adversarial learning. It has been proved that the generated samples perfectly fit the real data distribution when $P_g = P_{\text{data}}$. When the training reaches the set number of epochs or the losses converges, the training stops, and the data distribution of normal time series is captured.

Then the anomaly score GDScore is calculated based on the performance metric of the generator and the discriminator, where the generator metric is calculated by cosine similarity between original samples and generated samples. Finally, the threshold τ is automatically determined by the minimum value of GDSorce obtained from the training, which is used to detect outliers. The details are described in Section III-D.

In the testing stage, latent vectors $Z_{\text{test}} = \{Z^j_{\text{test}}, j = 1, 2, \ldots, n\}$ encoded by testing samples and the testing samples X_{test} are input into the model to obtain the generated samples $G(Z_{\text{test}})$ from the generator and the outputs $D(X_{\text{test}})$ and $D(G(Z_{\text{test}}))$ from the discriminator. Then, the anomaly score GDScore is calculated for each data point. One will be detected to be an outlier if its GDScore is higher than the threshold τ.

C. Architecture of Generator and Discriminator

To simultaneously capture the temporal features and spatial features in a telemetry subsequence, CNN and LSTM are used as the network structure of the generator and the discriminator.

The convolutional layer of a CNN has a strong ability to extract local features. The first input feature map of CNN is the subsequence $X_{\{s_w\}}$ with size $(N, s_w, 1)$. Then, given the input feature map with size (h, w, C_{in}), where h and w are the height and width of the feature map, C_{in} is the number of channels, the output of the k^{th} convolutional layer C_{out_k} can be described as follows:

$$C_{\text{out}_k} = \sum_{\alpha=1}^{C_{\text{in}}} w_{(k,\alpha)} * \text{input}\left(C^{\alpha}_{\text{in}}\right) + b_{\alpha} \quad (3)$$

where $w_{(k,\alpha)}$ is the weight matrix of the α^{th} input channel and the k^{th} output channel, b_{α} is the bias of the α^{th} input channel, and $*$ denotes the convolution operation.

This article chooses LSTM as the network structure behind CNN for its ability to process the time series. Besides, since memory gates exist in LSTM, there is no need to calculate the impact of all historical data on the current time point, so the calculation can be greatly simplified.

The specific network structure of the generator and the discriminator is illustrated in Fig. 3. In the generator, the input is a set of random vectors Z of latent space, where Z_{train} denotes vectors in the training stage and Z_{test} denotes ones in the testing stage. Z is concatenated to a fully connected layer F_1 and reshaped into a 2-D matrix $X'_{\{s_w\}}$. Then the time-scale convolutional layer C_1 and the spatial-scale convolutional layer C_2 are concatenated. In C_1, the filter's receptive field on the time axis is set to 5 and the receptive field on the variable axis is set to 2. C_1 explores time span to achieve the optimal contextual dependencies, while at the same time minimizing the influence of correlation information between variables.

On the contrary, the spatial-scale convolutional kernel of C_2 is set to 2×5, which is used to extract the correlations between variables while minimizing contextual dependencies. These two convolutional layers are designed to extract the time-scale and the spatial-scale features, respectively. Three upsampling layers $\{U_1, U_2, U_3\}$ are followed by F_1, C_1, and C_2, which enlarge the feature maps. After a reshape operation, two LSTM layers $\{L_1, L_2\}$ and one fully connected layer F_2, $X'_{\{s_w\}}$ is transformed into a 2-D matrix $\hat{X}_{\{s_w\}}$.

In the discriminator, the input is the real-time series $X_{\{s_w\}}$ or the reconstructed time series $\hat{X}_{\{s_w\}}$. Same as the generator, the convolutional kernels of the two convolutional layers C_3, C_4 are set to 5×2 and 2×5, respectively. The input layer and two convolutional layers are followed by three max pooling layers $\{P_1, P_2, P_3\}$ to form a symmetric structure with the generator. After two LSTM layers $\{L_3, L_4\}$, one dense layer F_3 and the sigmoid activation function, the time series is converted to a number, which indicates the probability of whether the sample is identified correctly.

D. Anomaly Score–GDScore

Since the GAN-based model is chosen in our detection algorithm, the rule to determine an outlier should also be specifically designed according to the characteristics of this model. Therefore, we propose a metric, named GDScore, to judge whether a data point is an outlier by measuring the metrics of the generator and the discriminator. Generator metric G_m is calculated by cosine similarity between the original sample and the generated sample, which measures the reconstruction deviation of the generator. G_m can be calculated by

$$\begin{aligned} G_m &= \text{Cos i ne similarity}(X_{\text{test}}, G(Z_{\text{test}})) \\ &= \frac{\sum_{j=1}^{n} X_{\text{test}}^{j} \times G\left(Z_{\text{test}}^{j}\right)}{\sqrt{\sum_{j=1}^{n} \left(X_{\text{test}}^{j}\right)^{2}} \times \sqrt{\sum_{j=1}^{n} \left[G\left(Z_{\text{test}}^{j}\right)\right]^{2}}}. \end{aligned} \tag{4}$$

Discriminator metric D_m is measured using the predicted value output by the sigmoid function of the last layer, which directly represents the probability of being an outlier. D_m can be described as

$$D_m = \begin{cases} D\left(X_{\text{test}}^{j}\right), & X_{\text{test}}^{j} \in X_{\text{test}} \\ D\left(G\left(Z_{\text{test}}^{j}\right)\right), & G\left(Z_{\text{test}}^{j}\right) \in G\left(Z_{\text{test}}\right). \end{cases} \tag{5}$$

Then, the proposed GDScore is calculated by

$$\text{GDScore} = w_g(1 - G_m) + w_d(1 - D_m) \tag{6}$$

where $w_g \in [0, 1]$ and $w_d \in [0, 1]$ are the weights.

In (6), w_g and w_d are determined by the loss of generator and discriminator, respectively, after the operation on the training data set, which can be calculated by

$$w_g = \frac{D_{\text{loss}}}{G_{\text{loss}} + D_{\text{loss}}} \tag{7}$$

$$w_d = \frac{G_{\text{loss}}}{G_{\text{loss}} + D_{\text{loss}}} \tag{8}$$

where G_{loss} is the loss of generator and D_{loss} is the loss of discriminator. In the training stage, w_g is computed by the proportion of the discriminator loss in the total loss while w_d is the proportion of the generator loss in total loss. In the testing stage, w_g and w_d are used to calculated GDScore by (6).

Both G_{loss} and D_{loss} are calculated by the cross entropy loss. G_{loss} and D_{loss} can be expressed as

$$G_{\text{loss}} = -\log\left(1 - D\left(G\left(Z_{\text{train}}^{i}\right)\right)\right) \tag{9}$$

$$\begin{aligned} D_{\text{loss}} &= -\left[y \log D\left(X_{\text{test}}^{j}\right) + (1 - y)\log D\left(G\left(Z_{\text{test}}^{j}\right)\right)\right] \\ y &= \begin{cases} 1, & \text{if the input sample } j \in X_{\text{test}} \\ 0, & \text{if the input sample } j \in G(Z_{\text{test}}). \end{cases} \end{aligned} \tag{10}$$

In order to improve the fault tolerance rate of the generator and the discriminator in ST-GAN, the minimum value of the GDScore(X_{train}) is automatically set as the threshold τ

$$\tau = \min(\text{GDScore}(X_{\text{train}})). \tag{11}$$

Some published anomaly detection algorithms used peaks over threshold (POT) algorithm to mine the threshold τ [14]. POT is a threshold mining method using the second theory of extreme value, which assumes that the peak value in the time series satisfies the generalized Patorley distribution (GPD). However, this method may not work when there are many outliers which do not satisfy GPD. The process of the ST-GAN algorithm is shown in Algorithm 1.

Algorithm 1 ST-GAN-Based Telemetry Data Anomaly Detection

```
In the training stage:
  For the $l^{th}$ epoch in the number of training epochs do
    Transform the latent vector to the generated samples
    in the generator: $Z_{train} => G(Z_{train})$;
    Calculate the output of discriminator:
    $X_{train} => D(X_{train}), G(Z_{train}) => D(G(Z_{train}))$;
    Update the model parameters of discriminator by
    minimizing $D_{loss}$;
    Update the model parameters of the generator by
    minimizing $G_{loss}$;
  End for
    Record the loss of generator and discriminator to give
    the weights of GDScore
    Record the minimum value of GDScore as the anomaly
    threshold $\tau$;
    Save the model parameters in the training stage;
In the testing stage:
    Obtain the output of generator and discriminator
    $G(Z_{test})$, $D(X_{test})$ and $D(G(Z_{test}))$ by the saved
    model parameters;
    Calculate the GDScore of the testing data points;
    Detect the outliers by comparing with the threshold $\tau$;
```

IV. Experiments

A. Data Sets and Performance Index

We conduct experiments on a real data set and two public data sets, in which Power System of a Rocket (PSR) data set

TABLE I

DETAILS OF DATA SETS

Dataset name	Number of channels	The total length of time series	Number of variables
PSR	1	27928	19
SMAP	55	429735	25
MSL	27	66709	55

is from the China Academy of Launch Vehicle Technology, and Soil Moisture Active Passive (SMAP) Satellite and Mars Science Laboratory (MSL) Rover are two public telemetry data sets from NASA [7].

PSR data set contains telemetry data of a power system in a rocket, which contains 19 variables and each with 27 928 data points. The SMAP and MSL data sets contain multiple channels, each with data from the same variables measured by different sensors. SMAP data set contains a time series of 25 variables and each variable has 429 735 data points of all the 55 channels. MSL data set contains 55 variables, and the total length of all the 27 channels is 66 709. Table I shows the details of the three data sets. For these data sets, the normal data points are labeled with 1 while outliers are labeled with 0. Then, the original time series is divided into the training set, the valid set, and the testing set. Note that the training set only contains normal time series.

We use commonly used metrics to evaluate the performance of our proposed ST-GAN algorithm, which are Precision, Recall, F1-score, and Run-time for each sample [15].

B. Experimental Settings and Platform

In the data preprocessing stage, we set the window size as $s_w = 50$ and step length as $s_t = 10$, then split the original time series into required subsequences.

In the generator, we set the length of the input latent vector Z to 100. In the network as shown in Fig. 3, the number of feature maps of the convolution layer C_1 with a kernel size of 5×2 and the convolution layer C_2 with a kernel size of 2×5 are both set to 64. The upsampling layers $\{U_1, U_2, U_3\}$ adopt the double expansion. In the discriminator, the two convolutional layers C_3, C_4 contain 32 and 64 channels, respectively. The kernels of the max pooling layers $\{P_1, P_2, P_3\}$ are both set to 2×2. The two-layer LSTM $\{L_1, L_2\}$ of the generator and the discriminator, respectively, contain 80 and 40 cells, and a dropout is added after each layer of LSTM to avoid overfitting. Besides, the ReLU activation function and batch normalization are used to avoid gradient disappearance and increase convergence speed. The details of layers are shown in Table II.

Both the generator and the discriminator use the Adam optimizer and the loss function of cross entropy. The batch size and number of epochs are set to 64 and 50, respectively. Because the discriminator converges faster than the generator during training, we set the initial learning rate of the generator to be higher, i.e., 0.002, and the 0.0001 for the discriminator.

TABLE II

DETAILS OF LAYERS IN PROPOSED ST-GAN

Layers	C_1	C_2	C_3	C_4	$\{P_1, P_2, P_3\}$	L_1	L_2
Kernel size	5×2	2×5	5×2	2×5	2×2	—	—
Channels	64	64	32	64	—	80	40

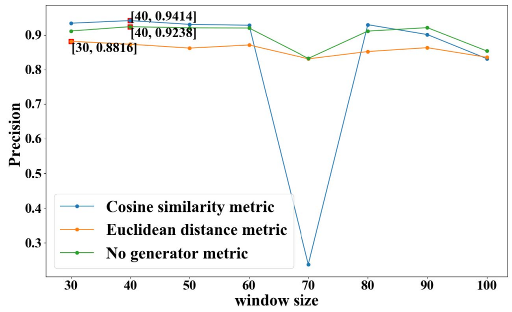

Fig. 4. Influence of window size and generator metric on the Precision.

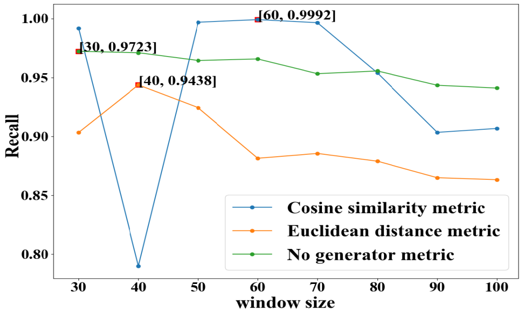

Fig. 5. Influence of window size and generator metric on the Recall.

The details of training parameters are shown in Table III. In order to obtain a reliable result and reduce randomness in the training phase, samples are trained and tested ten times, respectively, and then the standard deviations of performance indexes are calculated.

The proposed ST-GAN anomaly detection algorithm is implemented in Python 3.7 with the help of the deep learning library TensorFlow-GPU 1.8.0. Experiments are performed on a Huawei G5500 Linux server with NVIDIA Tesla V100 GPU.

C. Results and Comparisons

1) Setting of Window Size and Generator Metric: In order to set appropriate window size s_w and generator metric G_m, experiments are conducted on the validation data set. Take PSR data set as an illustrative example. The window size is set to $s_w = 30 + i \times 10, i = 0, 1, 2, \ldots, 7$ and the generator metric is set to the cosine similarity or the Euclidean distance [16]. Then, the Precision, Recall, and F1 on the PSR data set are shown in Figs. 4–6. By the figures, we can observe that the optimal value of Precision, Recall, and F1 both occur when cosine similarity is used as the generator metric. In addition, the results of cosine similarity metric fluctuate a lot, which shows that the window size has a great influence on the cosine

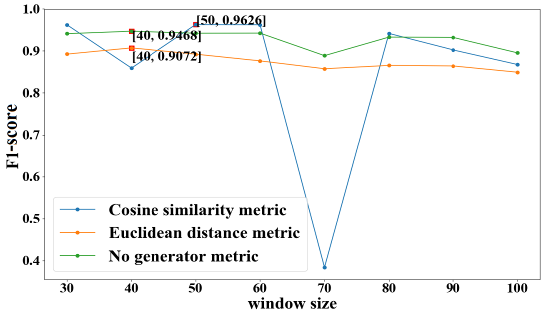

Fig. 6. Influence of window size and generator metric on the F1-score.

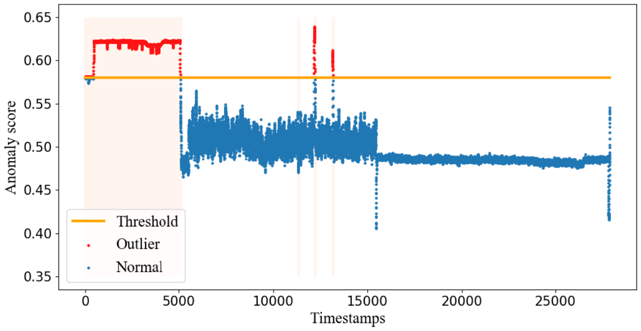

Fig. 7. Examples of anomaly detection results on PSR data set using ST-GAN.

TABLE III

TRAINING PARAMETERS OF PROPOSED ST-GAN

Generator optimizer	Discriminator optimizer	Loss function	Batch size	Epochs
Adam	Adam	Cross entropy	64	50

TABLE IV

PERFORMANCE OF FOUR METHODS

Dataset	Methods	Precision	Recall	F1	Run time for each sample (s)
PSR	ST-GAN	**92.75%**	**99.57%**	**96.04%**	0.00736
	LSTM-GAN	91.12%	90.25%	90.68%	0.00715
	GRU-VAE	87.22%	92.13%	89.61%	0.00822
	Iforest	82.11%	91.67%	86.63%	**0.00645**
SMAP	ST-GAN	**89.59%**	**99.58%**	**94.32%**	0.00979
	LSTM-GAN	88.95%	92.19%	90.54%	0.00925
	GRU-VAE	74.16%	97.76%	84.34%	0.01148
	Iforest	86.75%	89.59%	88.14%	**0.00892**
MSL	ST-GAN	87.78%	**99.36%**	**93.21%**	0.01371
	LSTM-GAN	**91.61%**	92.71%	92.16%	0.01245
	GRU-VAE	88.67%	91.17%	89.89%	0.01521
	Iforest	89.73%	90.26%	89.99%	**0.01167**

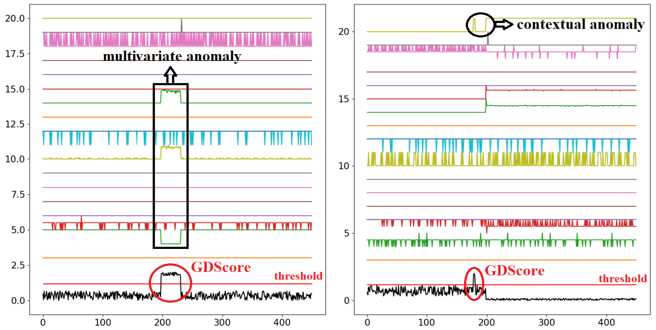

Fig. 8. Examples of detecting multivariate anomaly and contextual anomaly in the PSR data set using ST-GAN.

TABLE V

STANDARD DEVIATIONS OF TEN ROUNDS OF TRAINING ST-GAN

Dataset	Precision	Recall	F1
PSR	0.0059	0.0013	0.0036
SMAP	0.0050	0.0013	0.0029
MSL	0.0083	0.0021	0.0044

similarity metric. Therefore, the window size must be chosen carefully. In order to comprehensively evaluate the results of ST-GAN, we adopt the window size $s_w = 50$, when F1 reaches the optimal value of 96.26%.

2) Comparison Experiments: After setting the window size s_w to 50 and the generator metric G_m by cosine similarity, the result of the minimum GDScore can be obtained after the training phase, which sets as the value of anomaly threshold τ. Take PSR data set as an example, $\tau = 0.58$. Using this threshold, the outliers in PSR testing data set can be found. Examples are shown in Fig. 7. The detected outliers are mostly consistent with the actual outliers, indicating the accuracy of our proposed ST-GAN anomaly detection algorithm.

Besides, ST-GAN is able to detect multivariate anomalies caused by multiple variables and contextual anomalies under different situations, as mentioned in Fig. 1. The detection examples are shown in Fig. 8, in which a multivariate anomaly and a contextual anomaly have been correctly detected.

To demonstrate the effectiveness of our proposed algorithm, we compare its performance with LSTM-GAN which was used in our conference paper [1] and two other unsupervised anomaly detection algorithms, which are Iforest [9] and GRU-VAE [10]. The average values of Precision, Recall, and F1-score after ten rounds are calculated on the PSR, SMAP and MSL data sets, and the comparison results of these four algorithms are shown in Table IV. The standard deviations of Precision, Recall, and F1-score of ST-GAN after ten rounds of training and testing are shown in Table V.

As shown in Table IV, the result of F1-score by ST-GAN exceeds 90% on all data sets and is the highest among all algorithms for each data set. It demonstrates the satisfactory performance of our proposed ST-GAN algorithm and further, the better performance of this algorithm over others. As shown in Table IV, the standard deviations of Precision, Recall, and F1-score are all under 0.001, which proves that the proposed ST-GAN structure is stable for the telemetry data-based anomaly detection.

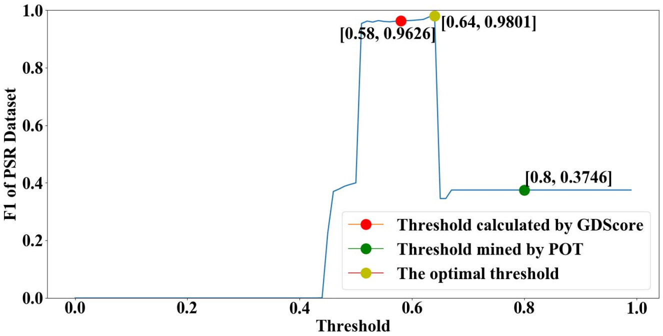

Fig. 9. Effect of threshold on PSR data set.

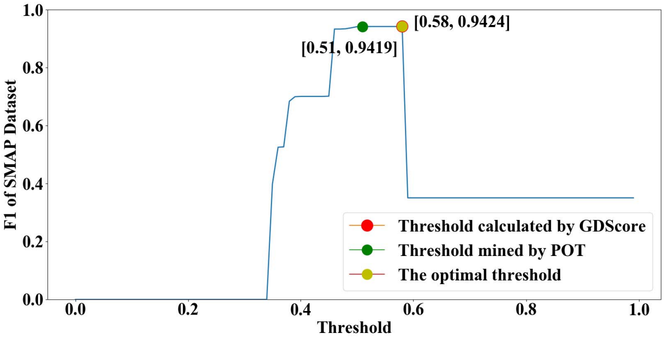

Fig. 10. Effect of threshold on SMAP data set.

As demonstrated in Table IV, the Precision, Recall, and F1-score of proposed ST-GAN on these three data sets are almost always better than the LSTM-GAN algorithm proposed in our conference paper [1], which demonstrates that the network structure combined CNN and LSTM is better than the LSTM structure. This might because the combination of spatial and temporal convolutional kernels can extract variable correlations and time features simultaneously, which offers more useful information in telemetry data for anomaly detection. Besides, it can be observed that the Precision of proposed ST-GAN on the PSR and SMAP data sets is higher than that of GRU-VAE and Iforest algorithms, and the Recall and F1-score obtained by proposed ST-GAN outperform that of these two algorithms on all PSR, SMAP and MSL data sets. Thus, the proposed ST-GAN has advantages over the other two methods on telemetry data-based anomaly detection. Moreover, the result of recall on all the three data sets using ST-GAN is above 99%, which indicates that ST-GAN has very few chances to give false alarms.

However, compared with Iforest, ST-GAN provides better performance at the expense of more run-time consumption. This is because Iforest does not require reconstruction of the telemetry data and ignores the time dependence between data points. Thus, it is only suitable for numerical anomalies, not for time-dependent anomalies. Compared with GRU-VAE, another data reconstruction-required method, ST-GAN saves more running time. Comprehensively speaking, the ST-GAN algorithm performs well in both effectiveness and applicability.

3) Influence of Threshold τ: The influence of threshold τ on the F1-score of the three data sets is considered. When the threshold is too small, Recall will be small, resulting in F1-score being too small; when the threshold is too large, Precision will be small, causing F1-score to be too small. So, only when the threshold is set appropriately, can F1-score be guaranteed to be optimal. Figs. 9–11 show the effect of threshold on the F1-score results of PSR, SMAP, and MSL testing data sets, respectively. They show that, with the increase of threshold τ, F1-score first increases to the peak and then decreases, finally converges to a steady value related to the proportion of outliers in the data sets. The optimal threshold values are 0.64, 0.58, and 0.43 for each data set. The minimum GDScore used for threshold is very close to the optimal threshold. It demonstrates our proposed rule to determine the anomaly threshold is effective.

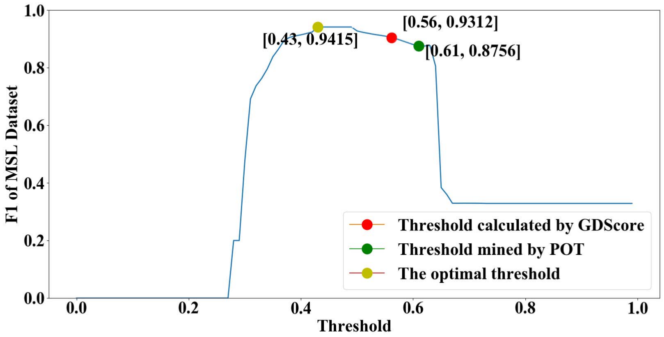

Fig. 11. Effect of threshold on MSL data set.

The threshold τ calculated by the minimum GDScore of the training data set is used to compare with the threshold τ_p mined by the POT method [14]. As shown in Figs. 9–11, the threshold τ for PSR data set is far closer to the optimal threshold compared with τ_p, while τ and τ_p of SMAP and MSL data sets are very close. This difference is probably because POT is more suitable to situations where very few extreme values (or outliers) exist. For the data set with a large proportion of outliers, such as PSR, POT method does not work well. From the comparison, we can observe that our proposed rule to determine the anomaly threshold is more effective and adaptable in real practice.

V. Conclusion

This article proposes an anomaly detection algorithm for telemetry data based on a spatial–temporal GAN. This ST-GAN algorithm develops a model based on GAN structure to learn the robust latent representation of normal time series. Considering time-related characteristics of a monitoring variable and complex correlation between multiple variables in telemetry data, this model considers a spatial–temporal structure for the generator and the discriminator in GAN, in which CNN and LSTM are combined. To determine whether an anomaly occurs, an anomaly score is also proposed, obtained by weighted fusion of the generator metric and the discriminator metric.

Experiments on three real telemetry data sets and comparison with three other anomaly detection algorithms demonstrate the effectiveness and advantages of our proposed algorithm.

Results show that our proposed algorithm has a better performance in finding the real anomalies while keeping a much lower false alarm rate. The experiments also demonstrate the anomaly threshold obtained by our proposed anomaly score GDScore is very close to the optimal threshold determined by ergodic search, and it is closer than other widely used methods. However, experiments also show the expense of computation becomes more as the network structure goes to more complex. Therefore, increasing the computation efficiency of ST-GAN is worth investigating. Moreover, more GAN-based anomaly algorithms and anomaly detection algorithm for different kinds of anomalies (e.g., degradation, intermittent anomaly) are also good research topics.

References

[1] Y. Song, J. Yu, D. Tang, D. Han, and S. Wang, “Telemetry data-based spacecraft anomaly detection using generative adversarial networks,” in *Proc. Int. Conf. Sens., Meas. Data Anal. Artif. Intell. (ICSMD)*, Xi’an, China, Oct. 2020, pp. 297–301.

[2] Y. He, Y. Peng, S. Wang, and D. Liu, “ADMOST: UAV flight data anomaly detection and mitigation via online subspace tracking,” *IEEE Trans. Instrum. Meas.*, vol. 68, no. 4, pp. 1035–1044, Apr. 2019.

[3] C. Chang, W. Nallo, R. Rastogi, D. Beugless, F. Mickey, and A. Shoop, “Satellite diagnostic system: An expert system for intelsat satellite operations,” in *Proc. IVth Eur. Aerosp. Conf. (EAC)*, 1992, pp. 321–327.

[4] Y. Li, E. Pan, and Y. Xiao, “On autoregressive model selection for the exponentially weighted moving average control chart of residuals in monitoring the mean of autocorrelated processes,” *Qual. Rel. Eng. Int.*, vol. 36, no. 7, pp. 2351–2369, Nov. 2020.

[5] L. Liu, Y. Peng, L. Wang, Y. Dong, D. Liu, and Q. Guo, “Improving EGT sensing data anomaly detection of aircraft auxiliary power unit,” *Chin. J. Aeronaut.*, vol. 33, no. 2, pp. 448–455, Feb. 2020.

[6] J. Wu, Z. Zhao, C. Sun, R. Yan, and X. Chen, “Fault-attention generative probabilistic adversarial autoencoder for machine anomaly detection,” *IEEE Trans. Ind. Informat.*, vol. 16, no. 12, pp. 7479–7488, Dec. 2020.

[7] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, “Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding,” in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 387–395.

[8] D. L. Iverson, “Inductive system health monitoring,” *Proc. Int. Conf. Artif. Intell.*, 2004, pp. 605–611.

[9] Y. Weng and L. Liu, “A sequence anomaly detection approach based on isolation forest algorithm for time-series,” in *High-Performance Computing Applications in Numerical Simulation and Edge Computing*. Singapore: Springer, 2019.

[10] Y. Su, “Robust anomaly detection for multivariate time series through stochastic recurrent neural network,” in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2019, pp. 2828–2837.

[11] I. J. Goodfellow *et al.*, “Generative adversarial nets,” in *Proc. NIPS*, 2014, pp. 1–8.

[12] D. Li, “MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks,” in *Proc. Int. Conf. Artif. Neural Netw.*, 2019, pp. 703–716.

[13] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar, “Efficient GAN-based anomaly detection,” 2018, *arXiv:1802.06222*. [Online]. Available: http://arxiv.org/abs/1802.06222

[14] A. Siffer, P.-A. Fouque, A. Termier, and C. Largouet, “Anomaly detection in streams with extreme value theory,” in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 1067–1075.

[15] C. Sammut and G. I. Webb, *Encyclopedia of Machine Learning and Data Mining*. Springer, 2017.

[16] R. Kalaivani and P. Lakshmi, “Adaptive neuro-fuzzy controller for vehicle suspension system,” in *Proc. 5th Int. Conf. Adv. Comput. (ICoAC)*, Dec. 2013, pp: 236-240.

[17] Y. Lyu, Z. Han, J. Zhong, C. Li, and Z. Liu, “A generic anomaly detection of catenary support components based on generative adversarial networks,” *IEEE Trans. Instrum. Meas.*, vol. 69, no. 5, pp. 2439–2448, May 2020.

Jinsong Yu (Member, IEEE) received the Ph.D. degree from the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China, in 2004.

From 2013 to 2014, he was a Visiting Scholar with the Department of Mechanical Engineering, University of Canterbury, Christchurch, Canterbury, New Zealand. He is currently an Associate Professor with the School of Automation Science and Electrical Engineering, Beihang University. His research interests include automatic testing and integrated vehicle health management.

Yue Song received the B.S. degree from North China Electric Power University, Baoding, China, in 2017, and the master’s degree from Beihang University, Beijing, China, in 2020, where she is currently pursuing the Ph.D. degree with the School of Automation Science and Electrical Engineering.

Her research interests include prognostic and health management, and deep learning-based anomaly detection.

Diyin Tang (Member, IEEE) received the B.S. and Ph.D. degrees from Beihang University, Beijing, China, in 2008 and 2015, respectively.

From 2012 to 2013, she was a Visiting Ph.D. Student with the Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, ON, Canada. She is currently an Assistant Professor with the School of Automation Science and Electrical Engineering, Beihang University. Her research interests include fault prognostics, degradation-based modeling, and condition-based maintenance.

Danyang Han received the bachelor’s degree from the Ocean University of China, Qingdao, China, in 2017. He is currently pursuing the Ph.D. degree with the School of Instrumentation Science and Opto-electronics Engineering, Beihang University, Beijing, China.

His research interest focuses on prognostic and health management technologies.

Jing Dai received the Ph.D. degree from the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China, in 2009.

He is currently a Senior Researcher at the China Academy of Launch Vehicle Technology Research and Development Center, Fengtai, Beijing. His research interests focus on aerospace vehicle design, and prognostic and health management.