applied sciences

MDPI

Article

Sparse Temporal Data-Driven SSA-CNN-LSTM-Based Fault Prediction of Electromechanical Equipment in Rail Transit Stations

Jing Xiong [1,2,*], Youchao Sun [1], Junzhou Sun [3], Yongbing Wan [4] and Gang Yu [3]

[1] College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China; sunyc@nuaa.edu.cn
[2] College of Air Transportation, Shanghai University of Engineering Science, Shanghai 201620, China
[3] SILC Business School, Shanghai University, Shanghai 201800, China; patricksun2001@163.com (J.S.); gyu@shu.edu.cn (G.Y.)
[4] Shanghai Rail Transit Technology Research Center, Shanghai 201103, China; wybingsh@163.com
* Correspondence: xjsh@sues.edu.cn; Tel.: +86-021-67791109

Abstract: Mechanical and electrical equipment is an important component of urban rail transit stations, and the service capacity of stations is affected by its reliability. To solve the problem of predicting faults in station mechanical and electrical equipment with sparse data, this study proposes a fault prediction framework based on SSA-CNN-LSTM. Firstly, this article proposes a fault enhancement method for station electromechanical equipment based on TimeGAN, which expands and generates data that conform to the temporal characteristics of the original dataset, to solve the problem of sparse data in the original fault dataset. An SSA-CNN-LSTM model is then established to extract effective data features from low-dimensional data with insufficient feature depth through structures such as convolutional layers and pooling layers in a CNN, determine the optimal hyperparameters, automatically optimize the model network size, solve the problem of the difficult determination of the neural network model size, and achieve accurate prediction of the fault rate of station electromechanical equipment. Finally, an engineering verification was conducted on the platform screen door (PSD) systems in stations on Shanghai Metro Lines 1, 5, 9, and 10. The experiments showed that the proposed prediction method improved the RMSE by 0.000699, the MAE by 0.00042, and the R2 index by 0.109779 when predicting the fault rate data of platform screen doors on all of the lines. When predicting the fault rate data of the screen doors on a single line, the performance of the model was better than that of the CNN-LSTM model optimized with the PSO algorithm.

Keywords: platform screen door system; fault prediction; sparse and weak feature data; data augmentation; CNN-LSTM

check for updates

Citation: Xiong, J.; Sun, Y.; Sun, J.; Wan, Y.; Yu, G. Sparse Temporal Data-Driven SSA-CNN-LSTM-Based Fault Prediction of Electromechanical Equipment in Rail Transit Stations. *Appl. Sci.* **2024**, *14*, 8156. https://doi.org/10.3390/app14188156

Academic Editor: Alessandro Lo Schiavo

Received: 27 June 2024
Revised: 31 August 2024
Accepted: 4 September 2024
Published: 11 September 2024

1. Introduction

1.1. Background

The electromechanical systems in stations are important parts of urban rail transit systems, as they provide services for the safe and orderly passage of passengers through stations. When station electromechanical equipment fails, the station service capacity declines, and serious faults can even induce safety accidents. Especially in the fully automatic operation mode, more equipment and interfaces are present in station electromechanical systems, which places higher requirements for the safe and stable operation of these systems. A platform screen door (PSD) system is critical equipment in modern metro engineering, and it is installed at the edge of the platform to protect the safety of passengers and prevent accidents caused by passengers falling from the platform [1,2]. A PSD system consists of two parts: the mechanical structure and electrical equipment. The mechanical part includes a door body structure and a door machine system, while the electrical part includes a monitoring system and a power supply system [3]. Platform screen door systems play a significant role in ensuring passengers' safety and the service level of rail transit operations. At present, the maintenance methods for PSD systems mainly include corrective maintenance, preventive maintenance, and predictive maintenance [4]. Corrective maintenance is a maintenance method in which assets are passively repaired after a fault, and maintenance requirements are determined through on-site inspection [5]. Preventive maintenance usually relies on a fixed schedule or mileage interval [6]. However, these two maintenance methods lack consideration of the actual operation of a station's electromechanical equipment, which may cause the system to reduce availability due to imperfect maintenance, thereby reducing its performance [7]. Predictive maintenance is based on characteristics that are predicted or known as a result of repeated analyses and the evaluation of important parameters of project degradation, and state-based maintenance is performed to extend the service life of the system [8,9]. It has become the main technical way to meet the needs of urban rail transit networks and automatic operation and to improve the level of operation service.

Predictive maintenance is a data-driven maintenance strategy that requires a large amount of high-quality data [10]. As PSD systems are an important component of station electromechanical equipment, fault data are only recorded when malfunctions occur. Existing research mainly uses the fault rate to achieve fault prediction for PSD systems. The original fault data only affect the calculation of the fault rate and do not affect the amount of fault rate data. The data volume of the fault rate is related to the time span of the original fault dataset, and it is a type of low-frequency data, making it difficult to meet the data requirements of predictive maintenance. In recent years, the SMOTE algorithm has been widely used for data augmentation; Wang et al. used the SMOTE method based on the Euclidean distance from the center to enhance the data in an access control terminal fault dataset [11]. Trinh and Kwon used the boundary SMOTE method to enhance data on the faults and residual life of machinery [12]. Duan et al. used the average radius SMOTE method to enhance a gear fault dataset [13]. The SMOTE method finds the nearest neighbors for each minority class sample and randomly interpolates between them to generate new samples, thereby achieving sample class balance before training the classifier [14]. However, the SMOTE method is mainly aimed at solving the problem of class imbalance [15]. The main problem of fault data enhancement of station electromechanical equipment is the generation of time-series data with temporal correlations that conform to the fault mode of platform screen doors, which is difficult to achieve with the SMOTE method. With the development of artificial neural network (ANN) technology, generative adversarial networks (GANs) based on recurrent networks have been used for data augmentation. Li et al. proposed an adaptive TSA-GAN for time-series prediction and verified it on the UCR 2015 time-series dataset [16]. TimeGAN proposed by Yoon et al. enables a GAN to capture the gradual dependence of time data by introducing a loss function [17]. Therefore, the use of a GAN for time-series optimization provides a new way to solve the problem of it being difficult to generate data that conform to the temporal distribution pattern of the original sequence when using traditional data enhancement methods.

In addition to a large amount of high-quality data, efficient and accurate prediction algorithms or models also have an important impact on the effectiveness of predictive maintenance. Traditional models, which are represented by autoregressive models and their variants, are mainly based on time-series models, which perform well in processing steady-state data [18]. This kind of model is mainly based on a linear autoregression of the time series itself. In real application scenarios, the time series obtained from data sources often have non-stationary and nonlinear characteristics, which limit the universality and accuracy of autoregressive models in practical applications. In order to improve the generalization and accuracy of prediction models, models based on neural networks (NNs) are widely used because of their strong learning and prediction abilities for inaccurate and nonlinear laws [19]. Guo et al. applied an error fusion of multiple sparse autoencoders

(EFMSAEs) to a time series of rolling bearings and predicted faults in mechanical systems through LSTM [20]. Guo et al. used an informer to predict electrical line-tripping faults [21]. Therefore, prediction models based on neural networks provide a new way to solve the problem of traditional prediction models having difficulties in generalizing and improving prediction accuracy.

1.2. Objectives and Scope

This study proposes an intelligent fault rate prediction framework for rail transit platform screen door systems. Firstly, a data enhancement method based on TimeGAN is adopted. Automatic coding components, including embedding and recovery functions, and adversarial components, including generators and discriminators, are jointly trained to learn sequence features, generated sequences, and cross-time iterations simultaneously. This method can solve the problem of sparse data in the original rail transit platform screen door fault dataset and generate augmented data that conform to the temporal characteristics of the original dataset. Secondly, a prediction model based on CNN-LSTM is used to extract effective data features from low-dimensional data with insufficient feature depth through a CNN, and shared convolution kernel parameters are used to reduce the model parameters. At the same time, the sparrow search algorithm (SSA) is used to optimize the key parameters of LSTM to solve the problem in which the scale of the neural network model is difficult to determine so as to improve the accuracy of the prediction of long-range dependent time series. Finally, the prediction framework mentioned above is applied to the fault rate prediction of platform screen door systems on four lines of the Shanghai rail transit system.

The rest of this article is organized as follows: Related work is described in Section 2. In Section 3, the proposed method for fault rate prediction in a rail transit PSD system is introduced. In Section 4, data from four lines of the Shanghai rail transit system are used in an engineering application to verify the effectiveness and accuracy. Finally, the conclusion and prospects are summarized in Section 5.

2. Literature Review

2.1. Data Augmentation Technology

Data augmentation technology can be divided into traditional methods of simple integration or adjustment of data and methods based on deep learning. Traditional enhancement methods include algorithms for the transformation and amplification of the data themselves and fusion methods for multiple and heterogeneous types of data to fuse data from different sources. Conversion and amplification algorithms for the data themselves come from the field of image recognition, in which different types of conversion are performed on the data, such as clipping, scaling, or translation [22–24]. However, due to the particularity of time-series data distributions, such algorithms cannot be directly applied for the enhancement of time-series data. Improved basic data enhancement methods for time series include jitter, arrangement, etc. The jitter method is designed to take advantage of the noise in data and simulate it to generate new samples. Zha et al. used the jitter method to enhance time-series data on wind power generation, thereby extending a sparse and unbalanced original dataset that recorded the working state of wind power equipment. Arrangement is performed to generate new data by specifying a time window and rearranging data from it [25]. Sun et al. improved the arrangement method and proposed a dynamic time-warping algorithm to enhance test data on rolling bearings of electromechanical equipment [26]. However, the learning of jitter for noise requires one to adapt to a variety of situations; otherwise, it will lead to negative learning. The arrangement method has the problem of not retaining time dependence, which may cause invalid samples to be generated. Multi-source heterogeneous data fusion refers to the integration of data from multiple data sources, such as different devices, sensors, systems, and networks, to make up for the shortcomings of a single data source and obtain data with higher dimensions. Sun et al. proposed a multi-source heterogeneous data fusion model based on a CatBoost feature layer and data layer, which fused sensor measurement data, offline inspection data, and video monitoring data from electromechanical equipment in a production line [27]. However, most of the fault data for platform screen door systems come from manual inspection, and the data sources are limited, so it is difficult to carry out effective heterogeneous data fusion.

Data enhancement methods based on deep learning are mainly based on the variational autoencoder (VAE) and generative adversarial network (GAN). The variational autoencoder (VAE) is composed of two parts: an inference network and a generation network. It realizes the approximate reasoning of a complex distribution by minimizing the KL divergence between two distributions and realizes the generation of continuous and smooth data. Fan et al. used the VAE to enhance fault detection data from semiconductor wafer manufacturing equipment and used the augmented data for classification learning [28]. In practical applications, the data distribution of time series is usually complex and contains a variety of internal associations. However, the VAE often assumes that the data distribution is a Gaussian distribution, which may lead to its inability to accurately capture the true distribution pattern of data in the process of data enhancement. The proposal of the generative adversarial network (GAN) solves the problem of it being difficult for generated data to simulate the real distribution pattern of the original data. Shi et al. used a GAN architecture to generate data from two different types of fault data sequences. The generator and discriminator of each GAN were composed of many-to-many LSTM models, so the model was able to process the input time information [29]. Sabir et al. augmented the data of DC signal samples by modifying the original convolution to create a one-dimensional deep convolution GAN (DCGAN), which solved the problem of it being difficult for the traditional GAN architecture to be directly used for time-series prediction [30]. Yoon et al. proposed TimeGAN, which adds a loss function to a GAN to capture the gradual dependence of data and solves the problem of the generator and discriminator of the traditional GAN model not accurately reproducing the time changes in the original data by using the recursive network [17].

2.2. Time-Series Prediction Based on Neural Networks

Because neural networks can automatically extract data features and have strong generalization abilities, they are widely used in the field of time-series prediction. At present, the mainstream neural network prediction methods can be divided into three categories in terms of their structure: the artificial neural network (ANN), convolutional neural network (CNN), and recurrent neural network (RNN), as well as a variant thereof: the long short-term memory network (LSTM).

ANN-based models are usually composed of input neurons, hidden neurons, and output neurons. Jain et al. used traditional techniques for trend reduction and seasonalization, and they used the results to train ANNs. The model was able to capture the nonlinear properties of complex time series and improve the prediction accuracy [31]. Moldovan and Buzugan used an ANN to predict the location and type of fault in faulty cables in a distribution system, and the prediction accuracy on a 20 kV distribution line dataset was 98% [32]. Compared with ANNs, CNNs have a stronger ability to extract relevant features due to the introduction of convolutional layers, which can effectively process multivariate time series [33]. Kashiparekh et al. proposed a pre-training model based on a CNN to classify time series. The model used a convolution filter to capture time features on multiple time scales and classified and predicted datasets that included electrical equipment and sensors [34]. Wu et al. proposed a convolutional neural network based on an adaptive adversarial network (ADACNN) and applied it to intelligent fault identification in the bearing mode for electromechanical equipment. Compared with a general CNN, it achieved a 4% accuracy advantage [35].

As a special neural network model, the recurrent structure of an RNN can capture the dependencies in time-series data. Che et al. proposed an RNN-based model to deal with multivariate time series with missing values. By using partial PSD system data at different time intervals and modeling them to capture the long-range time dependence, the prediction effect was improved [36]. Bezyan and Zmeureanu used an RNN to predict fault data in a ventilation machinery system and electromechanical heating system, respectively [37]. However, RNNs have problems such as gradient explosion and disappearance, and they can only remember short-term information when processing long sequences [38]. In order to solve the above problems, Hochreiter and Schmidhuber proposed a variant—the long short-term memory network (LSTM)—based on an RNN [39]. Xu et al. used Attention-LSTM, which was based on an attention mechanism, to predict the fault rate of mechanical equipment. Experiments on a bearing fault dataset showed that the model had high prediction accuracy [40]. However, due to the introduction of more parameters, LSTM has a problem where the network size is difficult to determine in practical applications [41]. Liu et al. used the PSO algorithm to optimize the LSTM parameters, solved the problem of the optimal scale of the LSTM model being difficult to determine, and predicted oil and gas faults in power grid transformer equipment [42].

2.3. Knowledge Gaps

The literature review reveals the following knowledge gaps:

(1) Data augmentation methods that transform and amplify the data themselves, such as jitter and arrangement, need to learn about the distribution pattern and noise of the data themselves in advance, and it is difficult for them to retain the time dependence of the original data. In addition, enhancement methods based on multi-source heterogeneous data fusion have certain requirements for the number and availability of data sources. Therefore, it is difficult to use traditional data enhancement methods for the enhancement of time-series data with a single data source and complex internal correlations, such as in fault rate prediction for electromechanical equipment.

(2) Although neural network models, especially LSTM, have been greatly optimized in terms of structural defects and have strong generalization capabilities and robustness, they have more hyperparameters, and the network scale is difficult to determine and optimize.

This study proposes a CNN-LSTM prediction framework based on TimeGAN data enhancement and optimized with the sparrow search algorithm (SSA) to predict the fault rate of platform screen door systems. Through TimeGAN, data that conform to the time characteristics of the original fault data are generated, and the fault dataset is augmented. The high-dimensional features of the fault sequence are extracted by the CNN, and LSTM is selected for fault rate prediction due to the need for long-term sequence prediction. The SSA is used to optimize the model parameters to achieve an accurate prediction of the fault rate of platform screen door systems.

3. Methodology

3.1. Fault Prediction Framework for Platform Screen Doors Based on SSA-CNN-LSTM

The prediction framework proposed in this study is shown in Figure 1, and it includes three main parts: data preprocessing, data enhancement, and fault rate prediction. In the data preprocessing stage, the fault rate and the average fault interval time are used as a fault evaluation index so as to objectively and accurately quantify the fault situation of the platform screen door system. After the component-level fault data records of the initial dataset are deduplicated and invalid features are cleaned, the calculation is performed according to the definition of the evaluation index, thereby converting the original dataset into system-level fault data containing the evaluation index.

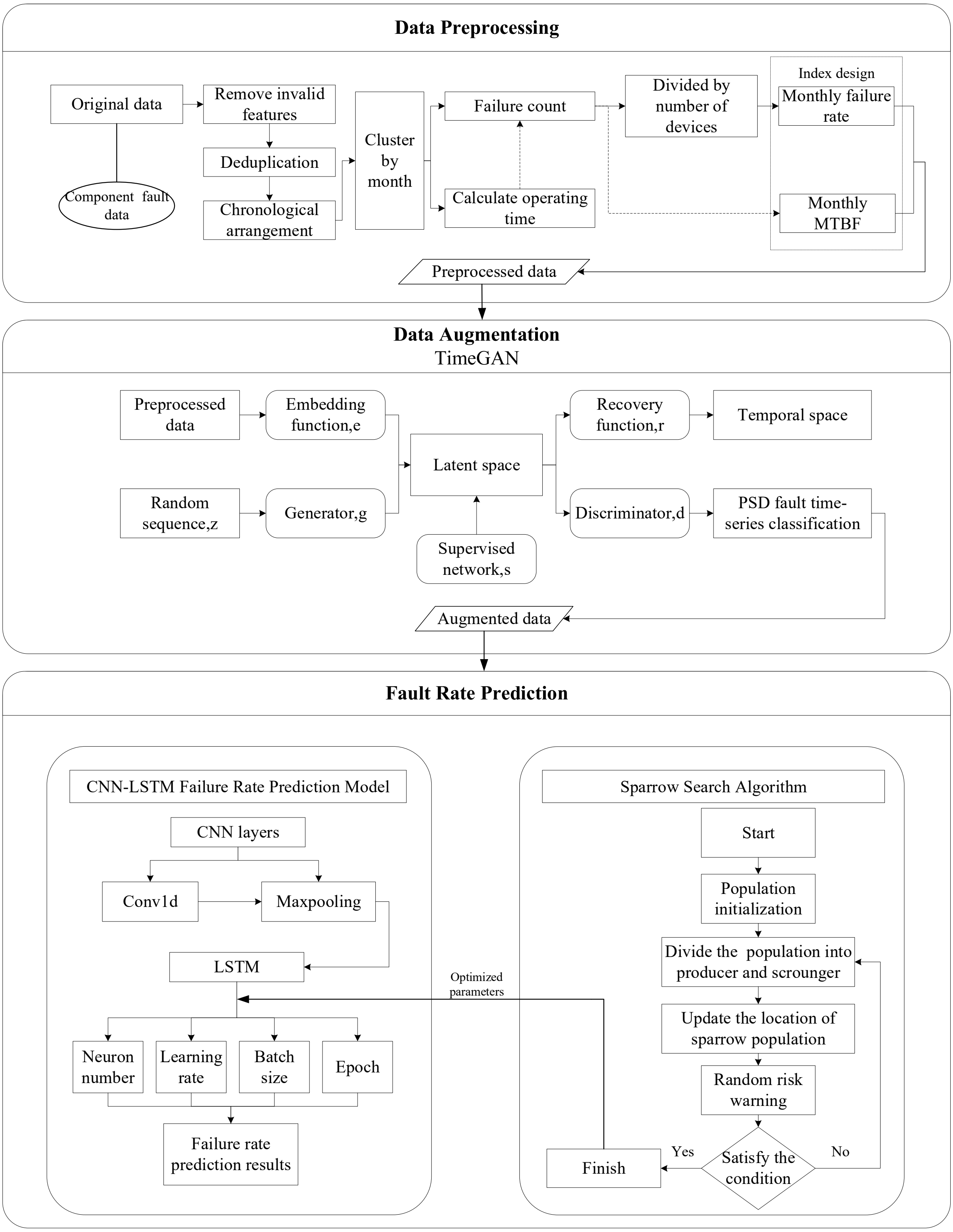


Figure 1. Fault prediction framework for platform screen doors based on SSA-CNN-LSTM.

The preprocessed data are augmented using TimeGAN, which learns the input platform screen door fault data through an automatic encoder network composed of an embedding function and a generating function, and it uses a supervised network to supervise and classify the original data and the generated data. Finally, the generated data are identified using an adversarial network composed of a recovery function and a discriminant function, and the generated data that conform to the spatial and temporal distribution of the original screen door fault data are given as output.

The proposed prediction model leverages a CNN to initially extract meaningful features from the augmented dataset. After the convolution layer, normalization layer, activation function layer, and pooling layer, the deep features in the original platform screen door fault data are extracted, which solves the problem of the depth of data features being insufficient and the difficulty in extracting effective features. Then, the sparrow search algorithm is used to determine the optimal parameters of LSTM and optimize its prediction effect. Finally, LSTM is used with the optimal parameters to predict the fault rate and improve the prediction accuracy.

3.2. Data Preprocessing

Table 1 shows the configuration of the original dataset for the training of the fault rate prediction model for platform screen door systems. The dataset includes fault time and space information that was automatically recorded by the Shanghai Rail Transit Command and Control Center, as well as fault records and maintenance records that were manually collected by the operation and maintenance personnel.

Table 1. Configuration of the raw dataset for predicting screen door system faults.

Data Source		**Data Information**
Monitoring Data	Temporal Information	Fault occurrence time Fault station
Maintenance Data	Fault Logging	Fault content Malfunction equipment Fault mode Fault cause Door function status
	Maintenance Record	Maintenance content Response time Handling time Processing response time Processing delay time

3.2.1. Design of Fault Assessment Indicators

Faults in PSD systems are usually caused by faults in the door machine system, faults in the control system, or other external factors, such as the intrusion of a foreign body. In order to quantitatively describe the reliability of the platform screen doors in a rail transit station, in combination with the data contained in the original dataset, the fault rate and average fault interval time were selected as indicators to represent the reliability of screen doors.

The fault rate refers to the probability that a product that has not yet failed at time t fails within a unit of time after time t, which is recorded as $\lambda(t)$. For finite samples, the number of samples is N, $n(t)$ samples fail after time t, and the fault number of the product at time $(t + \Delta t)$ is $n(t + \Delta t)$; then, the estimated fault rate is

$$\lambda(t) = \frac{n(t + \Delta t) - n(t)}{N \Delta t} \tag{1}$$

After a fault in a subway platform screen door, the faulty module is replaced, and the replaced module is returned to the factory for maintenance, which is equivalent to an update of the overall sample after maintenance. In this study, the fault rate in the next cycle (unit time) is approximately equal to the number of faults divided by the total number of samples.

The mean time between faults (MTBF) refers to the average time of normal operation between two faults. It is an important index for reflecting the reliability of equipment. The calculation formula is

$$MTBF = \frac{Equipment\ operating\ time \times Quantity}{Fault\ times} \tag{2}$$

3.2.2. Data Processing

Since the original dataset is a component-level fault record, it is necessary to process the component-level fault data into system-level fault data according to the above indicators. The specific data processing flow is shown in Figure 2, and it includes cleaning invalid features, deduplication, sorting, conditional clustering and counting, and time interval calculation.

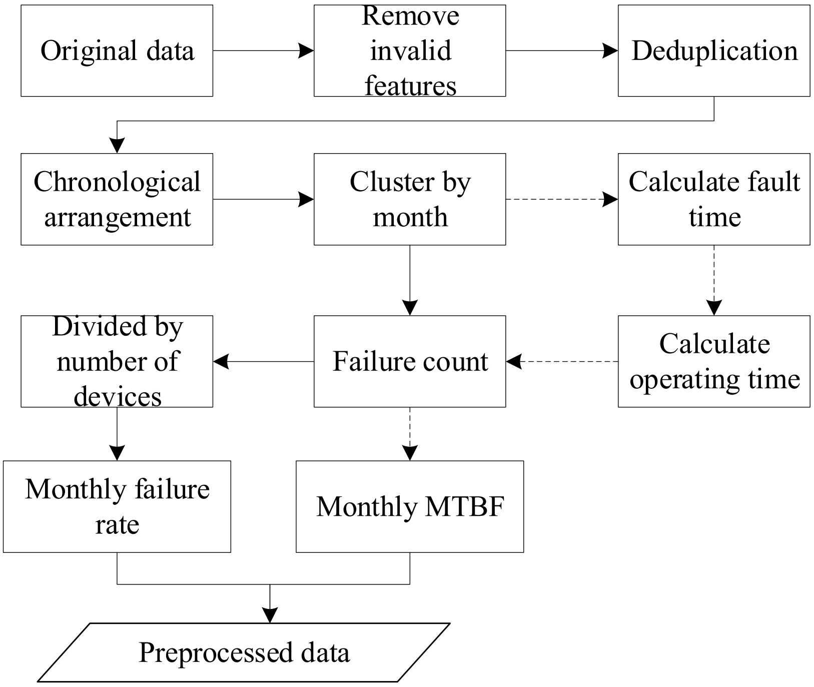


Figure 2. Data preprocessing flowchart.

The original dataset includes dimensions such as the occurrence time, site, fault device, fault content, fault type, work content, and response time. With reference to the determined indicators, redundant dimensions such as work content and response time are removed, and feature dimensions such as occurrence time and site are retained. After that, the fault time is deduplicated according to the time, and the first fault record is retained. Then, the data after deduplication are sorted in chronological order to obtain the preliminary processed data. The preliminary processed data are clustered by year and month, and then, the monthly faults are counted to obtain the number thereof. Then, according to the fault rate calculation formula, the monthly fault rate can be obtained by dividing the total number of faults. On the other hand, the fault interval time is calculated using seconds for the preliminarily processed data, and the monthly average is calculated. Finally, the monthly average fault interval days are converted, and the processed dataset is formed together with the monthly fault rate obtained through the previous processing.

3.3. Data Augmentation

During the operation of a platform screen door, many factors, such as the door machine system and the external environment, affect its normal operation. These factors are usually nonlinear and dynamic, so it is difficult to establish a clear and accurate mathematical model. In addition, the platform screen door fault dataset also has specific temporal characteristics; the fault and maintenance situation of the current month affect the fault data of the next month. If the fault rate of the current month is high and most of the equipment is maintained, the fault rate of the next month will decrease, the MTBF will increase, and the overall fault rate will increase with time. To address the issue of data sparsity, TimeGAN is employed to identify and learn the significant statistical features derived from historical data. This approach can be used to understand the intricate structures and characteristics inherent in time-series data. Furthermore, it generates novel time series by capturing the distribution of a collection of random noise vectors while preserving the correlations present within the time series itself, that is, the relationship between the generated data and the preprocessed data, such as the autocorrelation between the monthly fault rate and the monthly average fault interval time, as well as the change characteristics of the fault rate over time.

3.3.1. TimeGAN Construction

TimeGAN consists of an autoencoder network, a supervised network, and an adversarial network. The specific structure is shown in Figure 3.

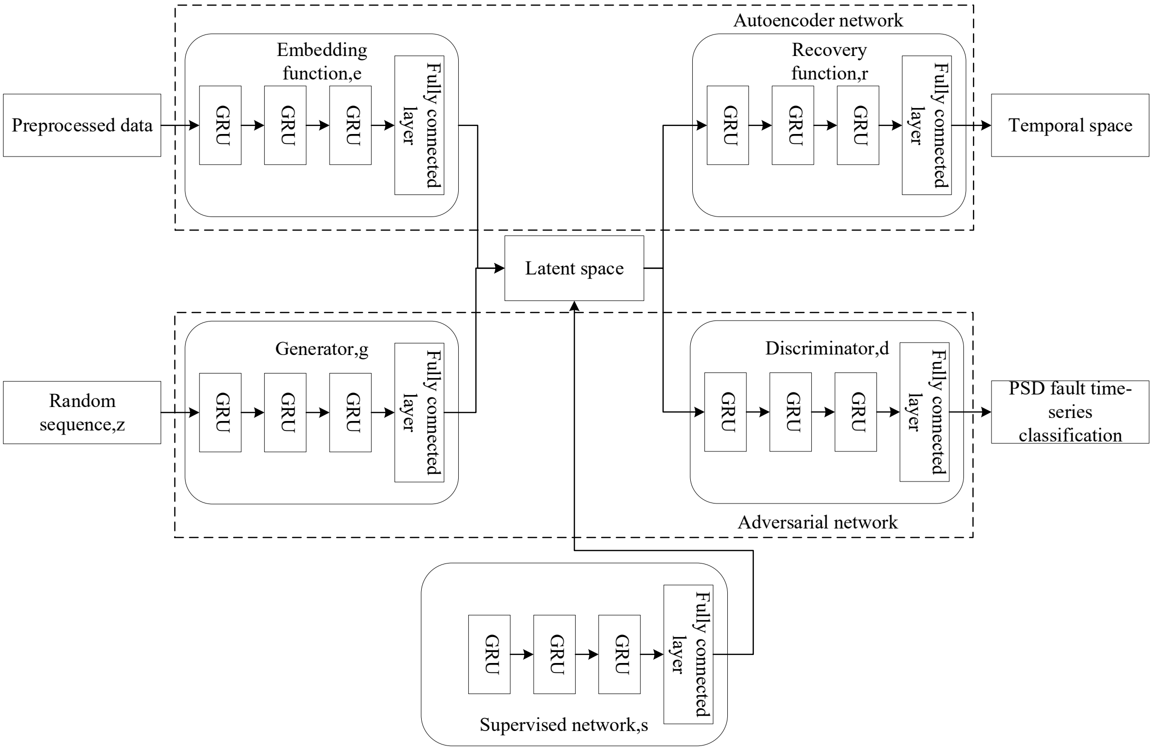


Figure 3. Fault data generation model for platform screen door systems based on TimeGAN.

- Autoencoder network construction

The embedding function and the recovery function establish a reversible mapping between the feature space and the latent space, thereby facilitating the adversarial network's capacity to learn the interrelationships inherent within time-series data. The embedding function is articulated as follows:

$$h_t = e_p(h_{t-1}, p_t) \tag{3}$$

In Formula (3), e_p denotes the embedding function, which comprises three layers of gated recurrent units (GRUs) followed by a fully connected layer. The hidden layer comprises 24 neurons, which utilize the sigmoid activation function. The variable p_t denotes the code corresponding to the processed fault rate data of PSDs in the potential space. The subscript t denotes temporal information. Let h_t represent a low-dimensional temporal feature at time t. The recovery function is characterized as follows:

$$\widetilde{p}_t = r_p(h_t) \tag{4}$$

$$\hat{p}_t = r_p(\hat{E}_{sup,t}) \tag{5}$$

In Formulas (4) and (5), r_p denotes the recovery function, which serves the purpose of reconstructing h_t with respect to the data $\widetilde{p}_t$, ensuring that it maintains the same dimensionality as the processed PSD fault rate data. It further re-establishes the supervisory space $\hat{E}_{sup,t}$ of the generated data to correspond with the data $\hat{p}_t$, maintaining dimensional congruence with the processed PSD fault rate data.

- Supervised network construction

The potential space coding h_t corresponding to the fault data of screen doors after processing and the fault data E_t generated by the generator are expressed in the supervised space by the following formulas:

$$\hat{h}_{sup,t} = s_p(h_t) \tag{6}$$

$$\hat{E}_{sup,t} = s_p(E_t) \tag{7}$$

where s_p represents the supervision function, which is composed of three layers of GRUs and a fully connected layer, and *sigmoid* is used as the activation function. $\hat{h}_{sup,t}$ and $\hat{E}_{sup,t}$ are the representations in the supervised space of h_t and E_t.

- Adversarial network construction

The data produced by the generator from the stochastic time series are initially generated within the latent space. The formula associated with the generator network is delineated as follows:

$$E_t = g_p(E_{t-1}, z_t) \tag{8}$$

In Formula (8), g_p denotes the generating function, which comprises a three-layer architecture of gated recurrent units (GRUs) in conjunction with a fully connected layer. Let E_t denote the PSD fault data generated by the generator at time t, while E_{t-1} signifies the data produced in the preceding time step. The variable z_t represents a stochastic time series that possesses the same dimensionality as the preprocessed PSD fault rate data. The discriminator operates within the potential space to differentiate between three specific domains: the potential space h_t representing the fault data generated by screen doors after input processing, the potential space E_t associated with the generation of fault data, and the supervised space $\hat{E}_{sup,t}$, which is also concerned with the generation of fault data. The formula pertinent to the discriminator network is delineated as follows:

$$c_{real,t} = d_p(h_t) \tag{9}$$

$$c_{fake\_e,t} = d_p(E_t) \tag{10}$$

$$c_{fake,t} = d_p(\hat{E}_{sup,t}) \tag{11}$$

In Formulas (9)–(11), d_p denotes the discriminant function, which exhibits a structural similarity to the generating function. The discriminant function utilizes the processed fault data alongside the generated data as input variables for classification purposes. In the presented formula, the variable t denotes the potential spatial classification outcome derived from the processed fault data. The term $c_{fake,t}$ signifies the supervised spatial classification result associated with the generated data, whereas $c_{fake\_e,t}$ refers to the potential spatial classification outcome of the generated data.

3.3.2. PSD Fault Data Augmentation Based on TimeGAN

The platform screen door fault data enhancement process based on TimeGAN is shown in Figure 4. The optimization of the generator loss, discriminator loss, and embedding function loss is accomplished through the application of gradient descent methods.

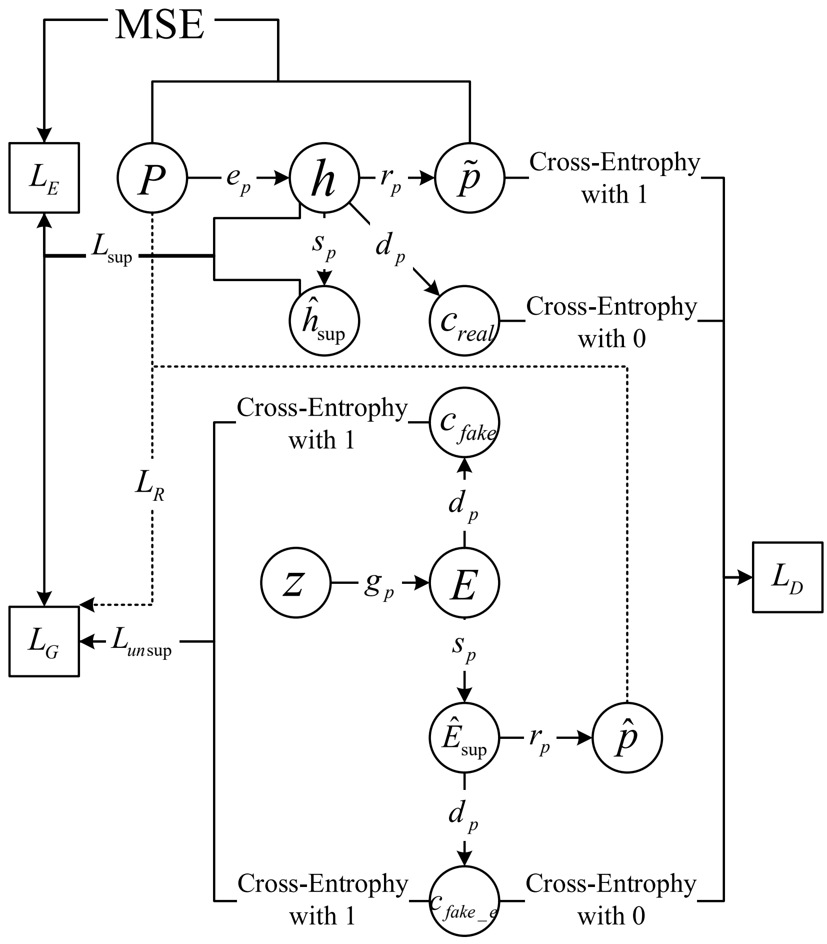


Figure 4. Fault data augmentation process for a PSD system based on TimeGAN.

- Generator loss L_G

The generator loss comprises three components: the unsupervised loss, denoted by L_{unsup}; the supervised loss, represented by L_{sup}; and the recovery loss, indicated as L_R. L_{unsup} denotes the cumulative cross-entropy between $c_{fake,t}$ and the value of 1, in addition to the cross-entropy between $c_{fake,t}$ and the value of 1. This relationship suggests that the data produced by the generator are discernible by the discriminator. L_{sup} denotes the root-mean-square error between the variables h_t and $\hat{h}_{sup,t}$. The underlying principle involves inputting the preprocessed fault data from the platform screen door system into the generator and subsequently calculating the error between the generated data and the initially provided input data. The recovery loss (L_R) quantifies the aggregate difference between the variance and the mean of the estimated values $\hat{p}_t$ and the actual values p_t. This indicates that the recovery function is capable of effectively reconstructing the data. The calculation of the generator loss (L_G) is conducted as follows [43]:

$$L_G = 0.1 \times L_{unsup} + 100\sqrt{L_{sup}} + 100 \times L_R \tag{12}$$

- Discriminator loss L_D

The discriminator loss L_D represents the sum of the cross-entropy of $c_{real,t}$ with 1, the cross-entropy of $c_{fake,t}$ with 0, and the cross-entropy of $c_{fake\_e,t}$ with 0, which indicates that the discriminator can accurately distinguish the generated data from the historical input data.

- Embedding function loss L_E

The embedding function loss L_E is composed of the supervised loss L_{sup} and the root-mean-square error of $\hat{p}_t$ and p_t. The calculation method is as follows [43]:

$$L_E = 10 \times \sqrt{MSE(p_t, \hat{p}_t)} + 0.1 \times L_{sup} \tag{13}$$

3.4. SSA-CNN-LSTM Fault Prediction Model

3.4.1. CNN-LSTM Model

Due to the small feature dimension of the system-level fault data of platform screen doors, these low-dimensional data will lead to difficulty in neural network feature extraction and affect the final prediction results. The architecture of the CNN-LSTM model proposed in this study is illustrated in Figure 5. The input data undergo a structured processing sequence, which includes a convolutional layer, a pooling layer, a long short-term memory (LSTM) layer, a flattening layer, and a fully connected layer, ultimately culminating in the final output. The CNN used in the model consists of one-dimensional convolutional (Conv 1D) layers and max pooling layers; Conv 1D is commonly used to process sequence data, including time-series data [44].

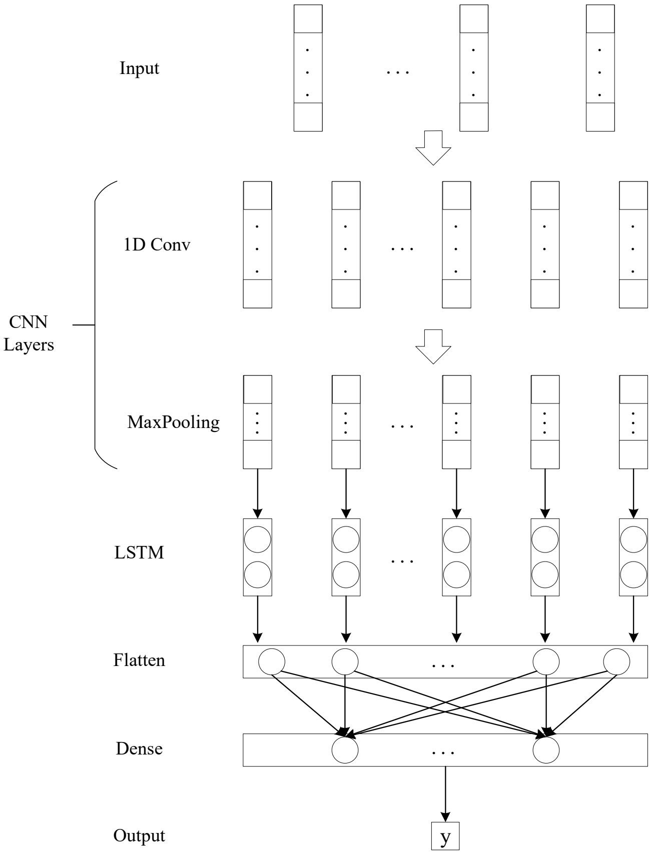


Figure 5. CNN-LSTM model structure.

The number of hyperparameters within a neural network significantly influences the accuracy of its results. Consequently, the sparrow search algorithm was chosen as a method for the automatic optimization of the hyperparameters with the aim of enhancing the predictive accuracy of the model.

3.4.2. SSA-CNN-LSTM Model

In the sparrow search algorithm, the update of the producer's location can be illustrated as shown in Formula (14):

$$r_{i,j}^{t+1} = \begin{cases} r_{i,j}^{t} \cdot e^{-\frac{i}{\alpha \cdot iter_{max}}} & if\ R_2 < ST \\ r_{i,j}^{t} + P \cdot L & if\ R_2 \geq ST \end{cases} \tag{14}$$

where t is the current epoch, $j = 1, 2, 3, \ldots, d$. $iter_{max}$ is the largest epoch number. $r_{i,j}$ denotes the sparrow's position information in dimension j at the i th iteration. $R_2 \in [0, 1]$ and $ST \in [0.5, 1]$ represent the warning value and the safety value, respectively. $\alpha \in (0, 1)$ is a random number. P is a random number subject to a normal distribution. L is a matrix containing only values of 1. When $R_2 < ST$, this indicates the absence of immediate danger, allowing the scrounger to conduct searches across an extensive area. When $R_2 \geq ST$, this signifies the presence of a potential hazard, prompting the issuance of an alarm. Subsequently, the affected population is relocated to a secure area to ensure their safety.

The update of the scrounger's location can be illustrated as shown in Formula (15):

$$r_{i,j}^{t+1} = \begin{cases} Q \cdot e^{\frac{r_{worst} - r_{i,j}^{t}}{\alpha \cdot iter_{max}}} & if\ i > \frac{n}{2} \\ r_p^{t+1} + \left| r_{i,j}^{t} - r_p^{t+1} \right| \cdot A^{+} \cdot L & otherwise \end{cases} \tag{15}$$

where r_p is the optimal position occupied by the producer, and r_{worst} is the current global worst position. A is a $1 \times d$ matrix, and each element is randomly assigned 1 or -1; $A^{+} = A^T \left(AA^T \right)^{-1}$. When $i > \frac{n}{2}$, this indicates that the i th scrounger with a lower fitness value is in a poor state and does not obtain food in a very hungry state.

In threatening situations, certain species of sparrows exhibit anti-predatory behaviors. The dynamics of their location updates can be illustrated as shown in Formula (16):

$$r_{i,j}^{t+1} = \begin{cases} r_{best}^{t} + \beta \cdot \left| r_{i,j}^{t} - r_{best}^{t} \right| & if\ f_i > f_g \\ r_{i,j}^{t} + K \cdot \left(\frac{\left| r_{i,j}^{t} - r_{worst}^{t} \right|}{(f_i - f_w + \varepsilon)} \right) & if\ f_i = f_g \end{cases} \tag{16}$$

where r_{best} denotes the present globally optimal position, and β acts as the step length's control factor, which adheres to a normal distribution of random numbers centered at 0 with a variance of 1. $K \in [-1, 1]$ represents a random number, while f_i signifies the current fitness value of an individual sparrow. f_w and f_g correspond to the worst and best fitness values. ε serves as a minimal constant, preventing a zero denominator. K represents the motion direction of the sparrow and is used as a step control parameter [45].

The sparrow search algorithm evaluates the fitness value at each population update to ascertain the classification accuracy of the algorithm. In this study, the fitness value is defined as the reciprocal of the classification accuracy. The measurement of the classification accuracy can be articulated in the following manner:

$$ACC = \frac{M}{N} \tag{17}$$

where M is the number of predicted values and actual values that are the same, N is the total number of samples, and the fitness function is as follows:

$$y = \frac{1}{ACC} \tag{18}$$

In this study, an SSA-CNN-LSTM model is constructed by combining the sparrow search algorithm and the CNN-LSTM model. The operating mechanism of the SSA-CNN-LSTM model is shown in Figure 6 and can be described as follows [41]:

1. Data preprocessing: Data annotation, dataset division, and data normalization.
2. SSA parameter initialization: The number of sparrows is set to n, the number of producers is set to PD, the number of sparrows perceiving danger is set to SD, the safety threshold is set to ST, and the alarm value is set to R_2.
3. The fitness value is calculated, and the locations of the producer and scrounger are updated.
4. According to anti-predatory behavior, the sparrow population's location is updated.
5. The data are input into the CNN, and the data pass through the CNN layer and the pooling layer.
6. The data enter the LSTM neural network and are input to the flattening layer and the fully connected layer through the LSTM layer.
7. Output results.

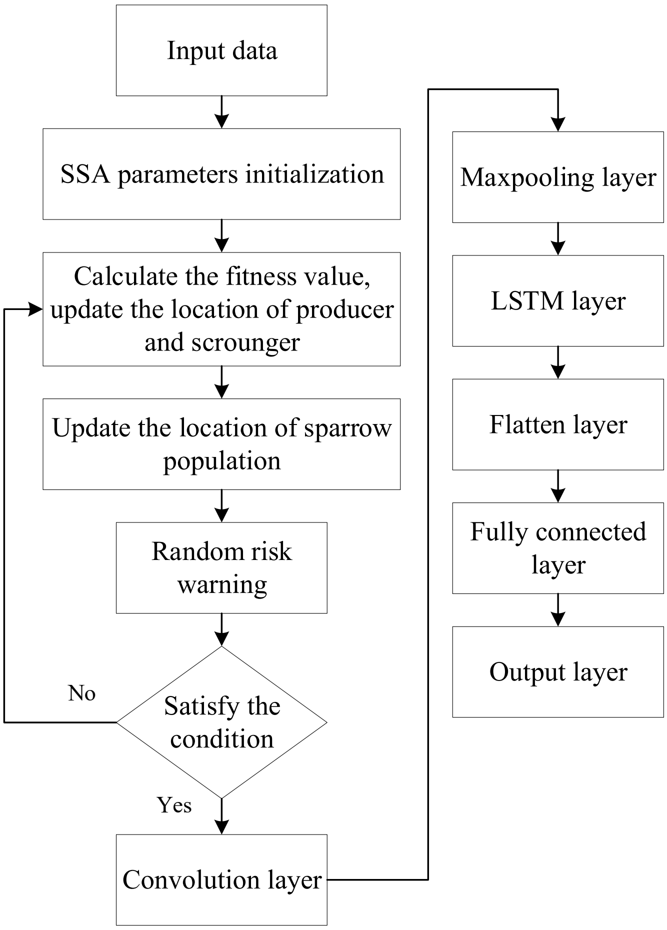

Input data
SSA parameters initialization
Calculate the fitness value, update the location of producer and scrounger
Update the location of sparrow population
Random risk warning
Satisfy the condition
No
Yes
Convolution layer
Maxpooling layer
LSTM layer
Flatten layer
Fully connected layer
Output layer

Figure 6. The operating mechanism of the SSA-CNN-LSTM model.

4. Case Study

As of December 2023, the Shanghai rail transit system had a total of 20 operating lines and 508 stations, with a total operating mileage of 831 km and a maximum daily passenger volume of 13.294 million. Since January 2020, to better formulate a maintenance strategy for the platform screen door systems, operators have begun to monitor faults in electromechanical equipment, such as station screen doors, through a command center in combination with manual inspection to better understand and prevent faults in the platform screen door systems.

Operators and managers seek to gain a precise understanding of the prospective fault trends associated with PSD systems. The application and verification processes employed in this study are illustrated in Figure 7. The processes encompassed three primary components: data preprocessing, data augmentation, and the prediction of fault rates.

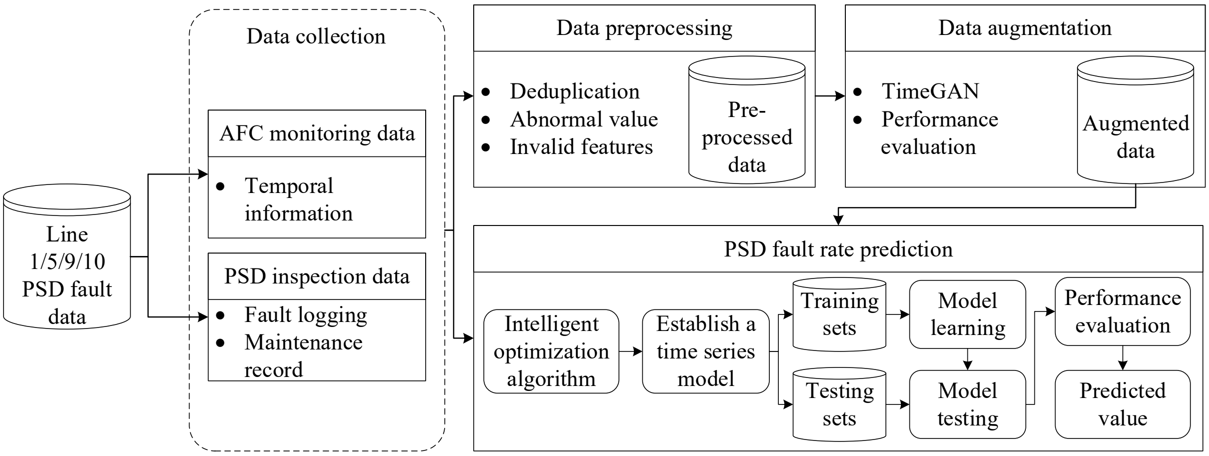


Figure 7. Fault rate prediction process for rail transit PSD system.

4.1. Data Preprocessing for Shanghai Rail Transit Lines

The platform screen door systems on Shanghai Rail Transit Lines 1, 5, 9, and 10 were taken as the research object, and 44 months of platform screen door fault data were collected from January 2020 to August 2023. Shanghai Metro Line 1 uses 6-car carriages with 5 doors per carriage, for a total of 28 stations and 1680 platform screen doors; Line 5 uses 6 carriages with 4 doors per carriage, totaling 19 stations and 912 platform screen doors. Line 9 uses 6 carriages with 4 doors per carriage, totaling 35 stations and 1680 platform screen doors; and Line 10 uses 6 carriages with 4 doors per carriage, totaling 37 stations and 1776 platform screen doors. The data were collected in the form of monitoring information from the command center and through manual inspection, and a total of 4708 sample data were collected. The configuration of data acquisition for screen door faults is shown in Table 2, and the original dataset is shown in Table 3. The original data were used to form a preprocessed dataset after cleaning invalid features, deduplication, sorting, conditional clustering and counting, and interval calculation. Due to the prediction target being the monthly fault rate of the platform screen door systems, the fault category or classification contained in each fault record had no significant correlation with the statistics and prediction of the monthly fault rate. Therefore, it was removed when cleaning up the dimensions of invalid data.

Table 2. Data configuration for PSD system faults on Shanghai Rail Transit Lines 1, 5, 9, and 10.

Categories of Data	**Parameters**		
Monitoring Information	1. Fault time	2. Fault station	
Inspection Information	3. Fault content 6. Fault cause 9. Response time 12. Delay time	4. Malfunctioning equipment 7. Door function status 10. Handling time	5. Fault mode 8. Maintenance content 11. Processing time

Table 3. Sample data of the original dataset.

Features	**Value**
Fault Serial Number	53020
Time	1 January 2020 12:41
Station	Line 9 Songjiang Sports Center Station
Malfunctioning Equipment	Platform screen door
Fault Content	1 January 2020 12:53:45 Line 9 AFC: Downward PSD fault alarm of Songjiang Sports Center

Table 3. *Cont.*

Features	Value
Maintenance Content	Replaced the drive power board, fixed
Response Time (min)	0.027
Handling Time (min)	2.45
Processing Time (min)	0.473
Delay Time (min)	21.869
Fault Cause	Drive power board
Fault Mode	Door machine system fault
Door Function Status	Cannot be operated

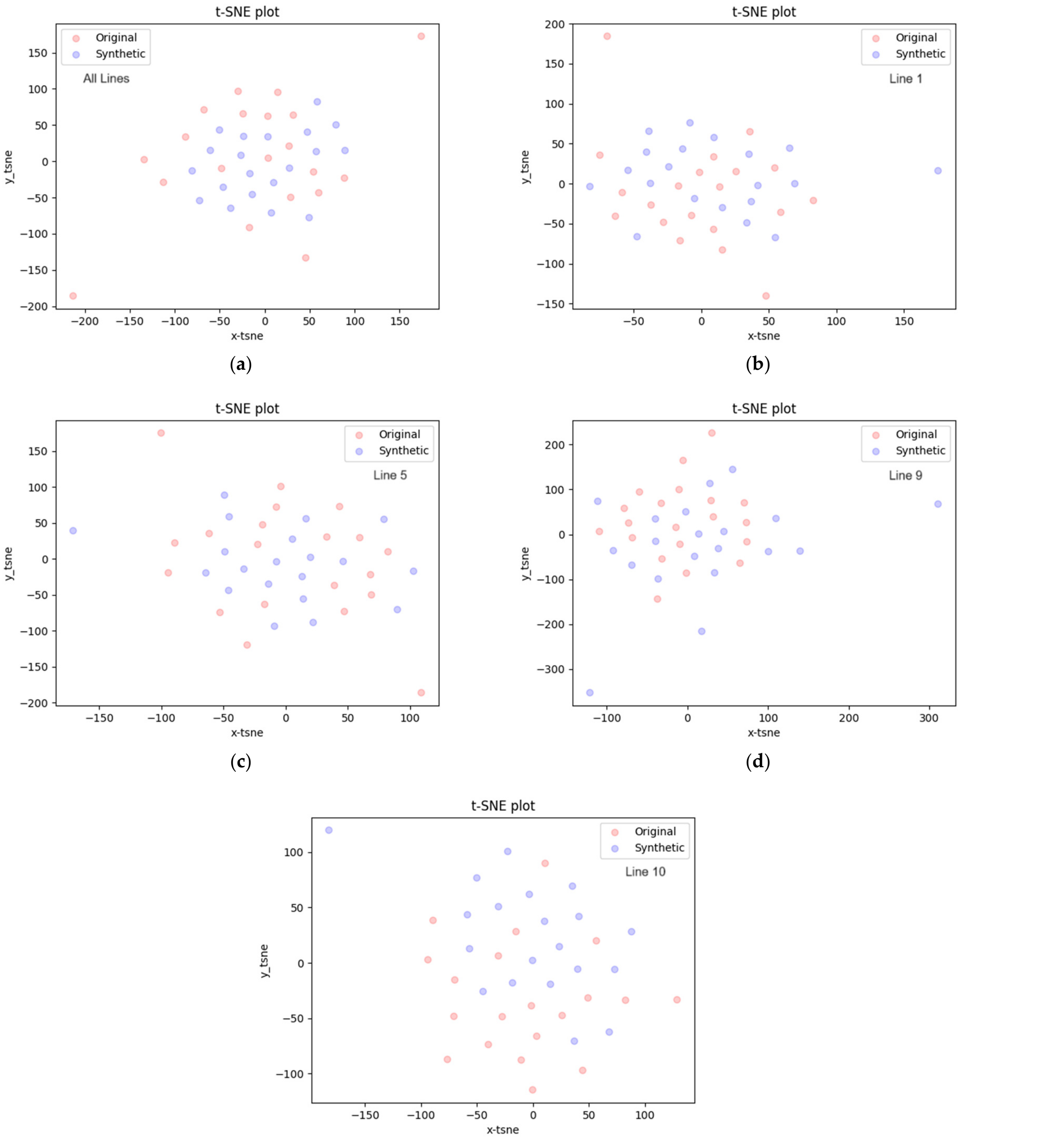
After preprocessing, the dataset (shown in Table 4) contained the monthly fault situation for 44 months, that is, the monthly fault rate and monthly MTBF data corresponding to each month from January 2020 to August 2023. In subsequent experiments, the entire processed dataset was input into the TimeGAN model for data augmentation.

Table 4. Sample data from the preprocessed dataset (data set of all lines).

Year and Month	Monthly Fault Rate	Monthly MTBF
January 2020	0.016203	1913.135
February 2020	0.005787	5011.192
March 2020	0.010912	2840.721
. . .	. . .	. . .
August 2023	0.011739	2640.666

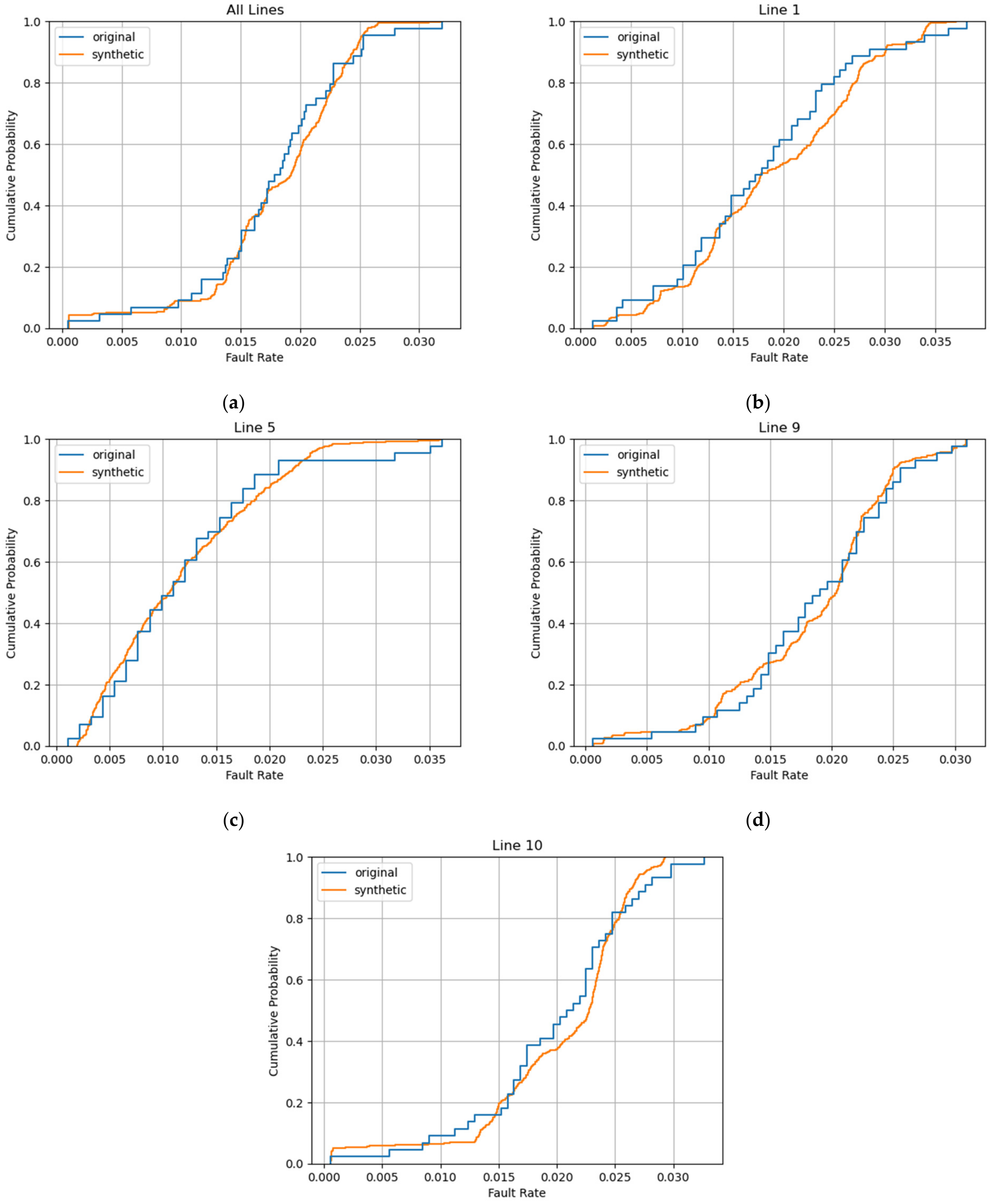
4.2. Data Augmentation for Shanghai Rail Transit Lines

According to the data enhancement method proposed in Section 3.3, first, the parameters of TimeGAN were debugged, a GRU was used as the generator, the hidden layers were set to 24, there were 3 layers, the batch size was set to 20, and the number of iterations was set to 10,000, with 24 units of data making up a group; then, the preprocessed data were input into the TimeGAN model to generate 20 groups of 480 units of data, including the monthly fault rate and monthly MTBF. This data augmentation operation was applied to the monthly fault data of all of the lines and the monthly fault data of each of the four lines, resulting in five datasets. The generated data were evaluated using t-SNE analysis, the cumulative probability distribution, and the discrimination score. The results of the t-SNE analysis are shown in Figure 8. It can be seen that the generated data in each dataset basically overlapped with the original data distribution area and had similar distribution characteristics, indicating that TimeGAN effectively learned and generated data that conformed to the original data distribution pattern. The cumulative probability distribution results are shown in Figure 9. It can be seen that the cumulative probability distribution curve of the generated data was able to basically fit the probability distribution curve of the original data, indicating that the generated data were able to better restore the probability distribution characteristics of the original data. The discrimination scores of the generated data are shown in Table 5, and the scores were all below 0.2, indicating that the classification error rate was low in each dataset, which further proved that the similarity between the generated data and the original data was close.



(**a**) (**b**) (**c**) (**d**) (**e**)

Figure 8. (**a**) The results of the t-SNE analysis of the dataset containing all lines; (**b**) the results of the t-SNE analysis of the Line 1 dataset; (**c**) the results of the t-SNE analysis of the Line 5 dataset; (**d**) the results of the t-SNE analysis of the Line 9 dataset; (**e**) the results of the t-SNE analysis of the Line 10 dataset.



(**a**)

(**b**)

(**c**)

(**d**)

(**e**)

Figure 9. (**a**) Results of the cumulative probability analysis of the dataset containing all lines; (**b**) results of the cumulative probability analysis of the Line 1 dataset; (**c**) results of the cumulative probability analysis of the Line 5 dataset; (**d**) results of the cumulative probability analysis of the Line 9 dataset; (**e**) results of the cumulative probability analysis of the Line 10 dataset.

Table 5. Discrimination scores of different datasets.

Data Sets	Discrimination Scores
All lines	0.155
Line 1	0.168
Line 5	0.162
Line 9	0.169
Line 10	0.181

4.3. Fault Prediction and Analysis

Prediction Results and Discussion

In accordance with the fault rate data prediction methodology and the augmented dataset delineated in Section 3.3, 80% of the augmented dataset was designated as the training set for the model, while the remaining 20% was allocated as the validation set [46]. The sliding window technique was employed to generate a fixed-step fault time series, which served as the input for the CNN-LSTM model. Subsequently, the parameters of the model were optimized using the shuffled shrinkage algorithm (SSA) to ascertain the optimal learning rate, the number of hidden neurons, the number of epochs, and the batch size for the LSTM model. The ultimate prediction results were derived through an iterative training process of the model, as illustrated in Table 6.

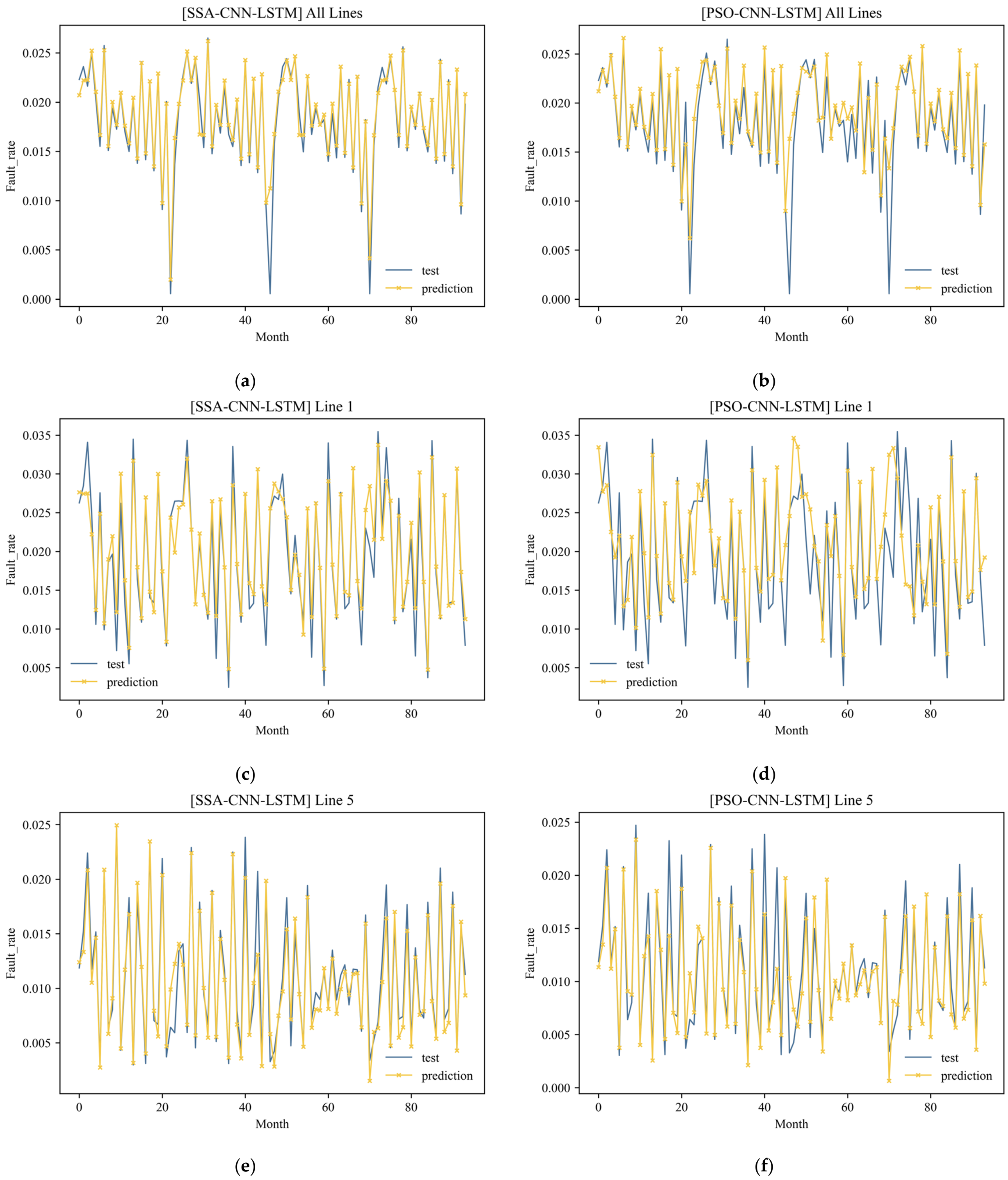
Table 6. Parameters of LSTM optimized using the SSA.

Parameters	Optimal Value				
	All Lines	Line 1	Line 5	Line 9	Line 10
Learning rate	0.009555	0.007819	0.001427	0.007261	0.006122
Epoch	93	171	151	250	357
Hidden neurons	32	27	44	39	38
Batch size	36	30	38	22	24

- Comparison of different optimization algorithms

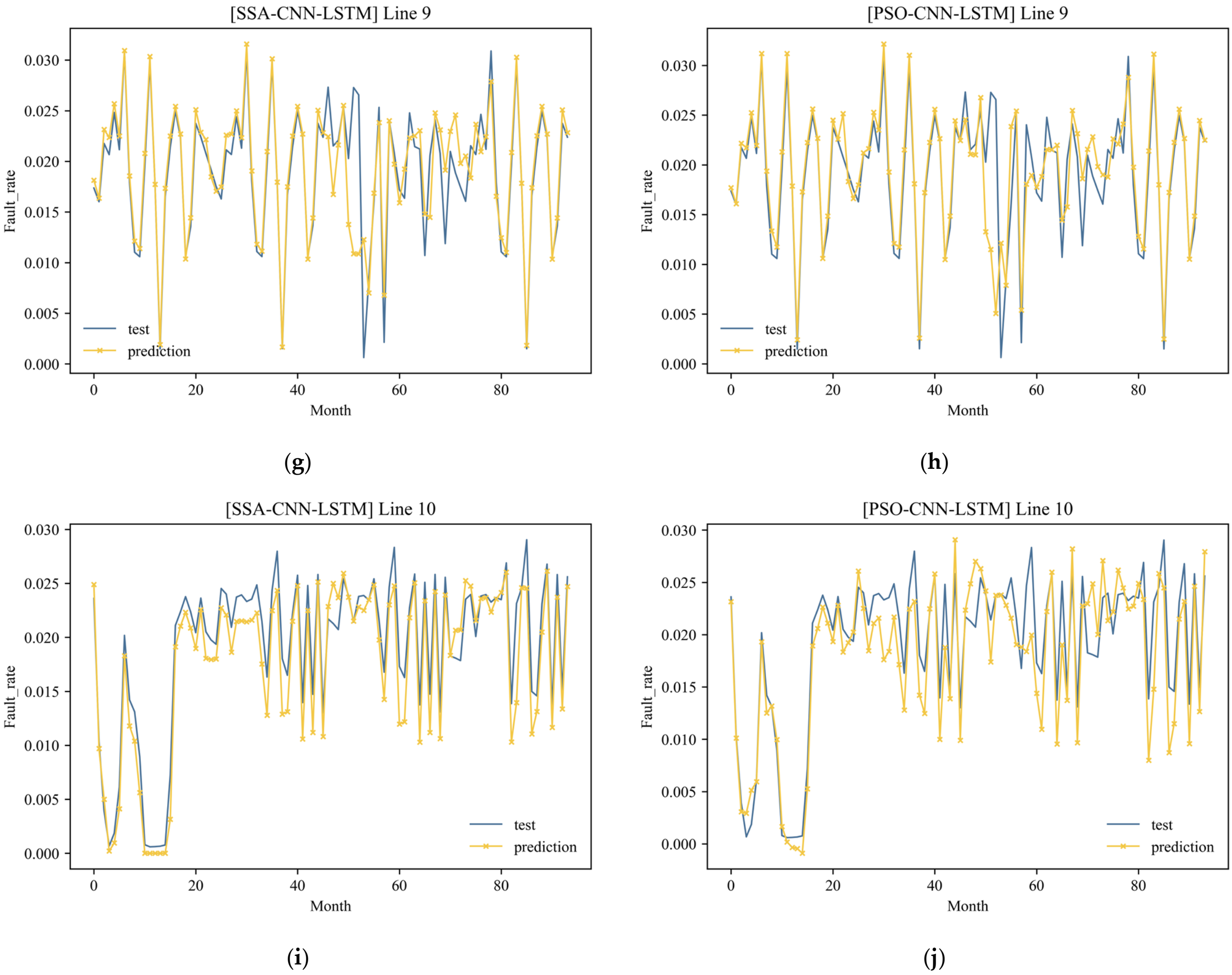
To improve the efficiency and prediction accuracy of neural networks, the particle swarm optimization (PSO) algorithm is used to effectively search for problems with a large solution space and find candidate solutions by simulating the motion of a bird swarm. It is an optimization algorithm that is commonly used in CNN-LSTM models [47].

To evaluate the prediction performance of the SSA-optimized CNN-LSTM model, the root-mean-square error (RMSE), mean absolute error (MAE), and determination coefficient (R2) were used. In this study, experiments were carried out on the augmented datasets containing all lines, Line 1, Line 5, Line 9, and Line 10. The performance of the SSA-CNN-LSTM and PSO-CNN-LSTM models is compared while using the same time window sequence and prediction step size as input, and the structure of the CNN-LSTM model was consistent. Figure 10 shows the prediction results of the SSA-CNN-LSTM and PSO-CNN-LSTM models for different datasets. It can be seen that the fitting degree of the SSA-CNN-LSTM prediction results was higher. Table 7 shows a performance comparison between the two, where the CNN-LSTM prediction model based on SSA optimization proposed in this study achieved a lower RMSE and MAE than those of traditional particle swarm optimization (PSO) on all datasets, and it had higher R2 scores. The RMSE is the square root of the average square of the difference between the predicted and actual values; the MAE is the average of the absolute sum of all prediction errors. From this, it can be seen that the smaller the values of the RMSE and MAE, the better the predictive performance of the model. This shows that the parameters of the CNN-LSTM model optimized using SSA were more reasonable, and the network scale was better, thus improving the prediction accuracy for the fault rate of platform screen doors.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 10. *Cont.*



(**g**)

(**h**)

(**i**)

(**j**)

Figure 10. (**a**) Prediction results for the PSD fault rate in the dataset containing all lines when using SSA-CNN-LSTM; (**b**) prediction results for the PSD fault rate in the dataset containing all lines when using PSO-CNN-LSTM; (**c**) prediction results for the PSD fault rate in the Line 1 dataset when using SSA-CNN-LSTM; (**d**) prediction results for the PSD fault rate in the Line 1 dataset when using PSO-CNN-LSTM; (**e**) prediction results for the PSD fault rate in Line 5 dataset when using SSA-CNN-LSTM; (**f**) prediction results for the PSD fault rate in the Line 5 dataset when using PSO-CNN-LSTM; (**g**) prediction results for the PSD fault rate in the Line 9 dataset when using SSA-CNN-LSTM; (**h**) prediction results for the PSD fault rate in the Line 9 dataset when using PSO-CNN-LSTM; (**i**) prediction results for the PSD fault rate in the Line 10 dataset when using SSA-CNN-LSTM; (**j**) prediction results for the PSD fault rate in the Line 10 dataset when using PSO-CNN-LSTM.

Table 7. Performance comparison of the CNN-LSTM model when using different optimization algorithms.

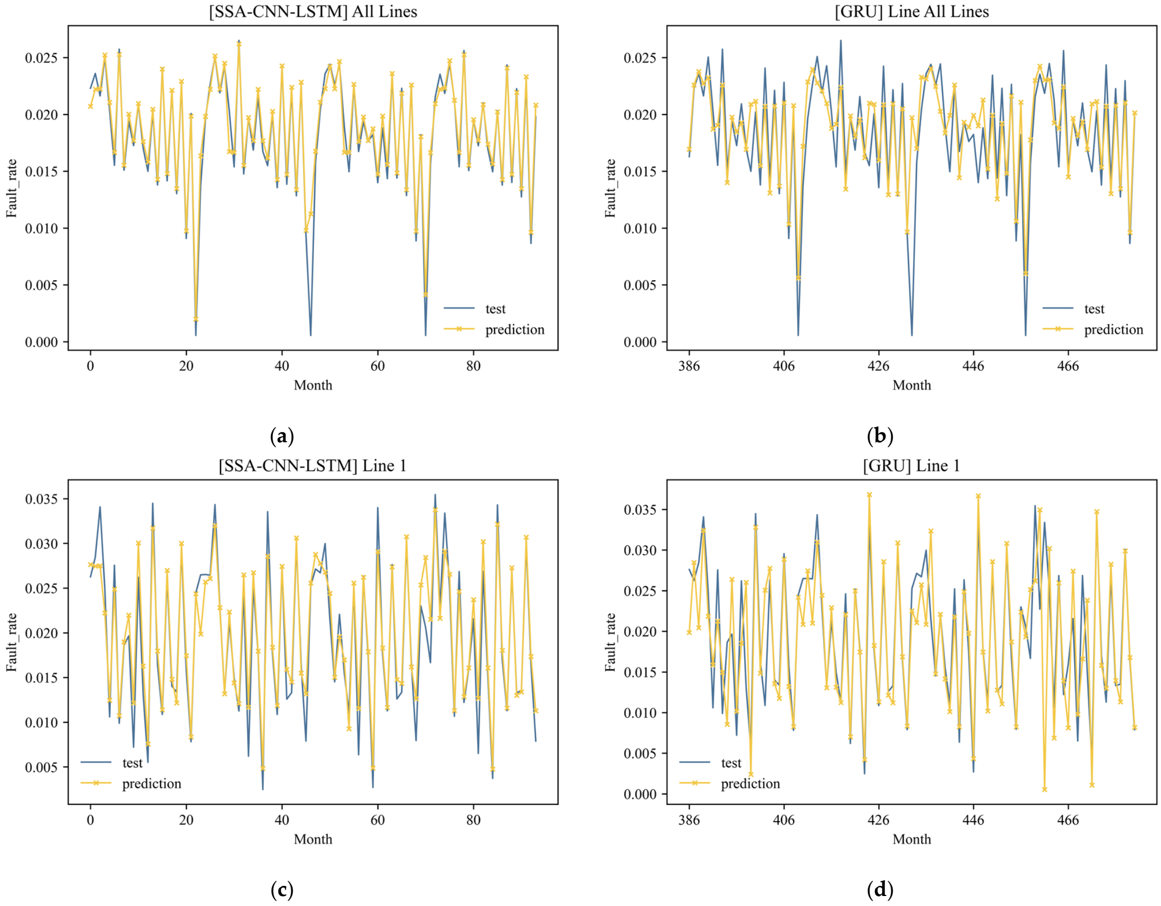
Criterion	**Algorithm**	**Optimal Value**				
		All Lines	**Line 1**	**Line 5**	**Line 9**	**Line 10**
RMSE	SSA	0.001445	0.002493	0.002232	0.003375	0.002724
	PSO	0.002636	0.003204	0.002564	0.003669	0.00322
MAE	SSA	0.000778	0.001771	0.00154	0.001917	0.002161
	PSO	0.001478	0.002487	0.001672	0.001991	0.00255
R2	SSA	0.92696	0.914977	0.882115	0.743619	0.866218
	PSO	0.756944	0.859563	0.844404	0.692574	0.804918

- Comparison of different prediction models

To verify the performance of the SSA-CNN-LSTM model, this study compared the prediction performance of SSA-CNN-LSTM and a gated recurrent unit (GRU) on the augmented datasets with all lines and with the four individual lines. Table 8 shows a comparison of the performance between the two. It can be seen in the table that the SSA-CNN-LSTM prediction model proposed in this study achieved lower RMSE and MAE values than those of the GRU on all datasets, and it had higher R2 scores. This showed that the SSA-CNN-LSTM model proposed in this study had better performance than that of the GRU in the face of long-range time-series prediction problems. Figure 11 shows the prediction results of SSA-CNN-LSTM and the GRU for the different datasets. It can be seen that the fitting degree of the SSA-CNN-LSTM prediction results was higher.

Table 8. Performance comparison between SSA-CNN-LSTM (SCL) and a GRU.

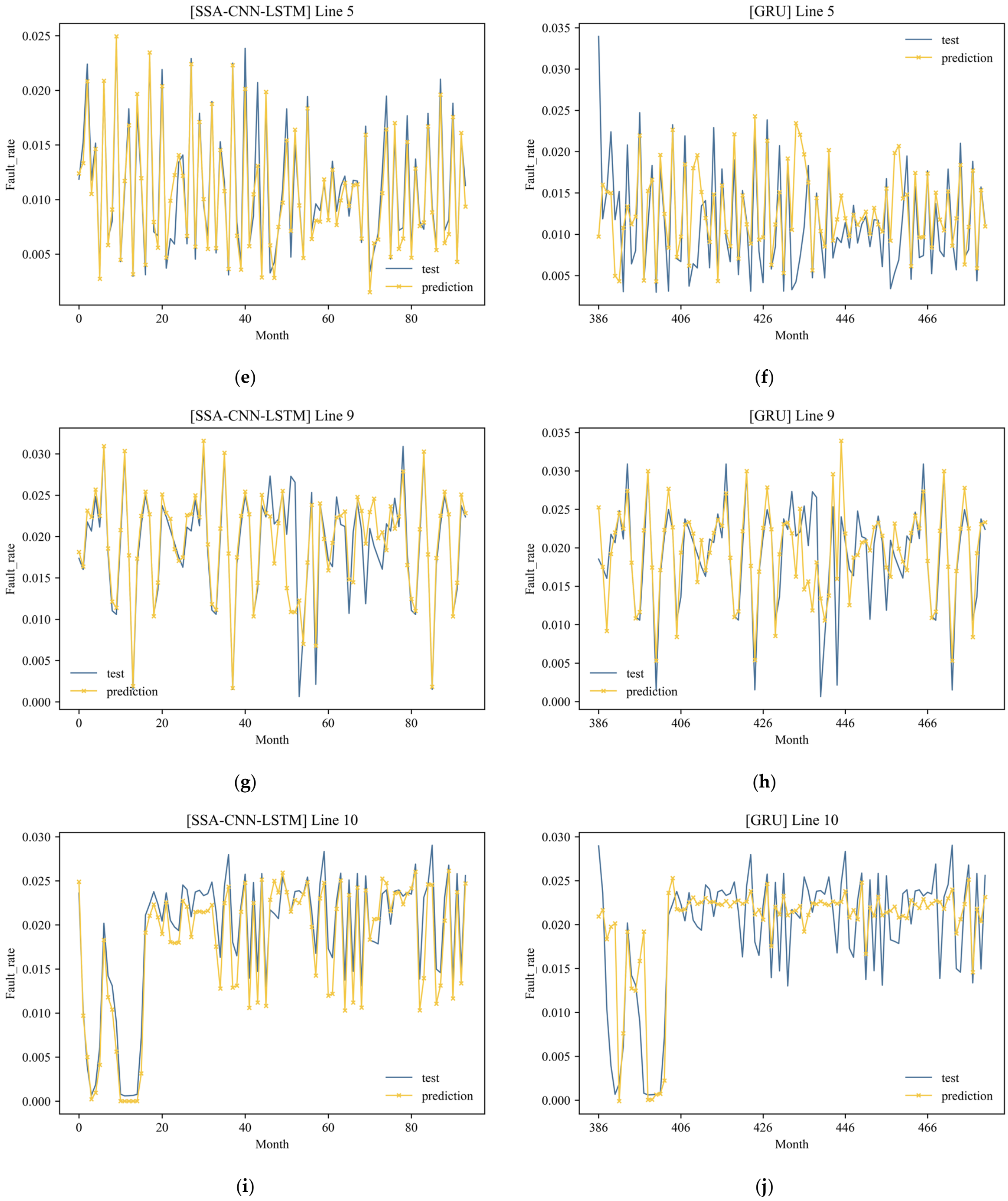
Criterion	**Model**	**Datasets (Augmented)**				
		All Lines	**Line 1**	**Line 5**	**Line 9**	**Line 10**
RMSE	SCL GRU	0.001897 0.003532	0.002638 0.003842	0.002232 0.004091	0.003375 0.003791	0.002724 0.004281
MAE	SCL GRU	0.000961 0.002305	0.001681 0.002257	0.00154 0.002722	0.001917 0.002598	0.002161 0.00326
R2	SSA GRU	0.874164 0.639592	0.904815 0.79047	0.882115 0.671733	0.739809 0.652535	0.860401 0.530113



(**a**) (**b**)

(**c**) (**d**)

Figure 11. *Cont.*



(**e**) (**f**)

(**g**) (**h**)

(**i**) (**j**)

Figure 11. (**a**) Prediction results for the PSD fault rate in the dataset containing all lines when using SSA-CNN-LSTM; (**b**) prediction results for the PSD fault rate in the dataset containing all lines when using the GRU; (**c**) prediction results for the PSD fault rate in the Line 1 dataset when using SSA-CNN-LSTM; (**d**) prediction results for the PSD fault rate in the Line 1 dataset when using the GRU; (**e**) prediction results for the PSD fault rate in the Line 5 dataset when using SSA-CNN-LSTM; (**f**) prediction results for the PSD fault rate in the Line 5 dataset when using the GRU; (**g**) prediction results for the PSD fault rate in the Line 9 dataset when using SSA-CNN-LSTM; (**h**) prediction results for the PSD fault rate in the Line 9 dataset when using the GRU; (**i**) prediction results for the PSD fault rate in the Line 10 dataset when using SSA-CNN-LSTM; (**j**) prediction results for the PSD fault rate in the Line 10 dataset when using the GRU.

- Ablation experiment

In order to verify the effectiveness of each component in the SSA-CNN-LSTM model, ablation experiments were designed and conducted. The performance of the model in removing the CNN layer, the performance when not using the SSA algorithm to optimize the parameters, and the performance when simultaneously removing the CNN layer and the SSA optimization algorithm were tested. The experimental results are shown in Tables 9–11. It can be seen that the prediction accuracy of the model with the CNN layer was greater than that of the model without the CNN layer, so the CNN layer had an important influence on the prediction accuracy of the model. In addition, the model using the SSA for intelligent optimization had higher prediction accuracy in each dataset than the traditional model that was based on experience for determining the parameters. The SSA-CNN-LSTM model with both the CNN layer and SSA for optimization achieved the best prediction accuracy in each dataset. Therefore, the results of the ablation experiments showed that both the CNN layer and the SSA for optimization in the SSA-CNN-LSTM model played an important role in the prediction accuracy of the model. Figures 12–16 show the prediction results of different methods in the ablation experiment on the fault rate of platform screen doors for each dataset, which were consistent with the conclusion of performance analysis. The poor performance of LSTM may have been due to the unclear distribution characteristics of the monthly fault rate data for certain lines, such as Line 5 and Line 10. The lack of a CNN layer to assist in feature extraction may have weakened the learning ability of LSTM for these lines' monthly fault rate data, resulting in a decrease in prediction accuracy.

Table 9. RMSE results of the ablation experiment on the SSA-CNN-LSTM model.

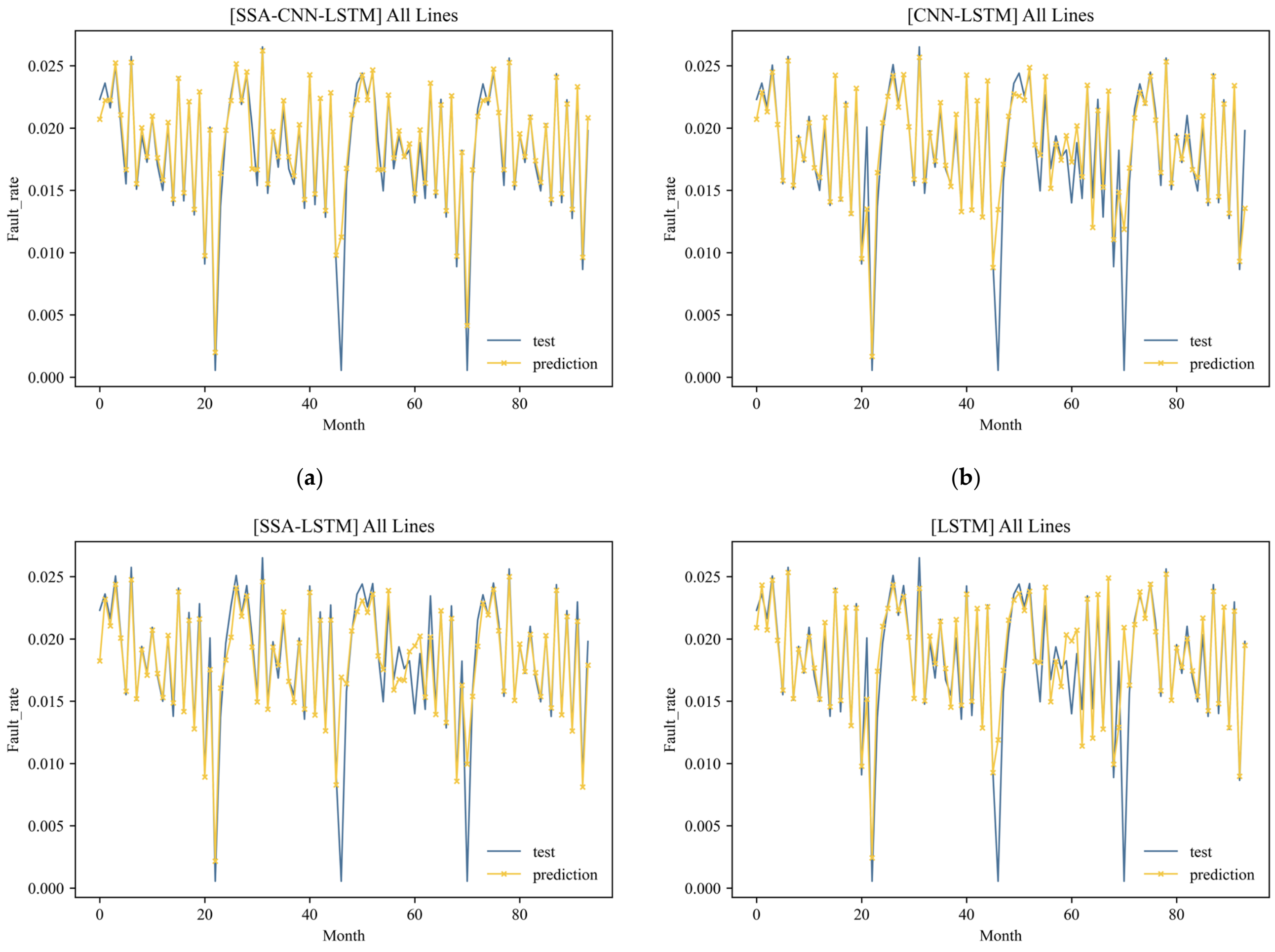
Methods	**RMSE**				
	All Lines	**Line 1**	**Line 5**	**Line 9**	**Line 10**
SSA-CNN-LSTM	0.001445	0.002493	0.002232	0.003375	0.002724
CNN-LSTM	0.00225	0.002711	0.00256	0.003542	0.00302
SSA-LSTM	0.002305	0.003932	0.003429	0.003949	0.003768
LSTM	0.002803	0.004864	0.004872	0.004057	0.005429

Table 10. MAE results of the ablation experiment on the SSA-CNN-LSTM model.

Methods	**MAE**				
	All Lines	**Line 1**	**Line 5**	**Line 9**	**Line 10**
SSA-CNN-LSTM	0.000778	0.001771	0.00154	0.001917	0.002161
CNN-LSTM	0.001103	0.002032	0.001689	0.002155	0.002506
SSA-LSTM	0.001092	0.002516	0.002176	0.002535	0.00287
LSTM	0.001271	0.003457	0.003423	0.00258	0.004082

Table 11. R2 results of the ablation experiment on the SSA-CNN-LSTM model.

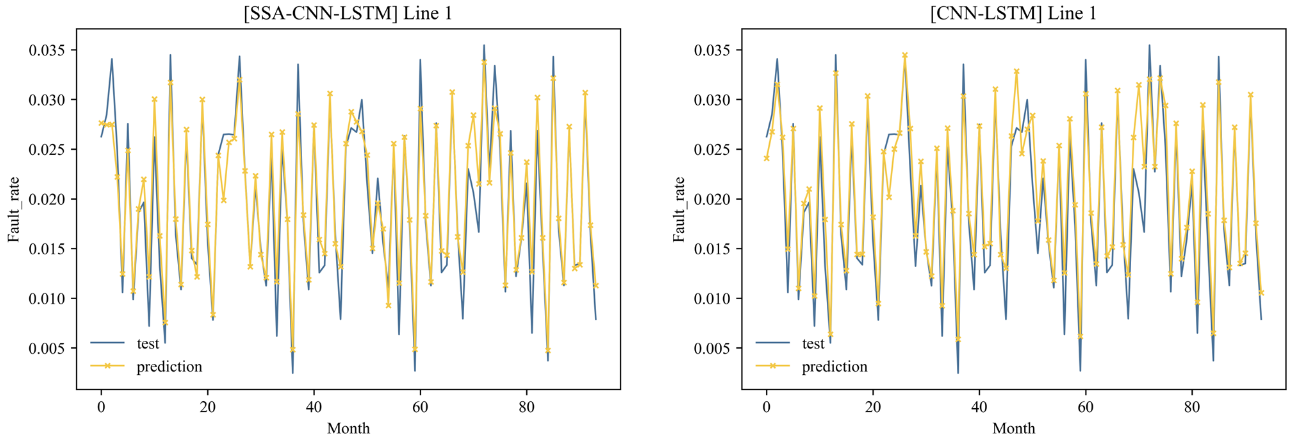
Methods	***R2***				
	All Lines	**Line 1**	**Line 5**	**Line 9**	**Line 10**
SSA-CNN-LSTM	0.92696	0.914977	0.882115	0.743619	0.866218
CNN-LSTM	0.823035	0.899422	0.844956	0.713404	0.828404
SSA-LSTM	0.81426	0.788502	0.721724	0.643882	0.732897
LSTM	0.725179	0.676328	0.43836	0.624026	0.445717



(**a**) (**b**)

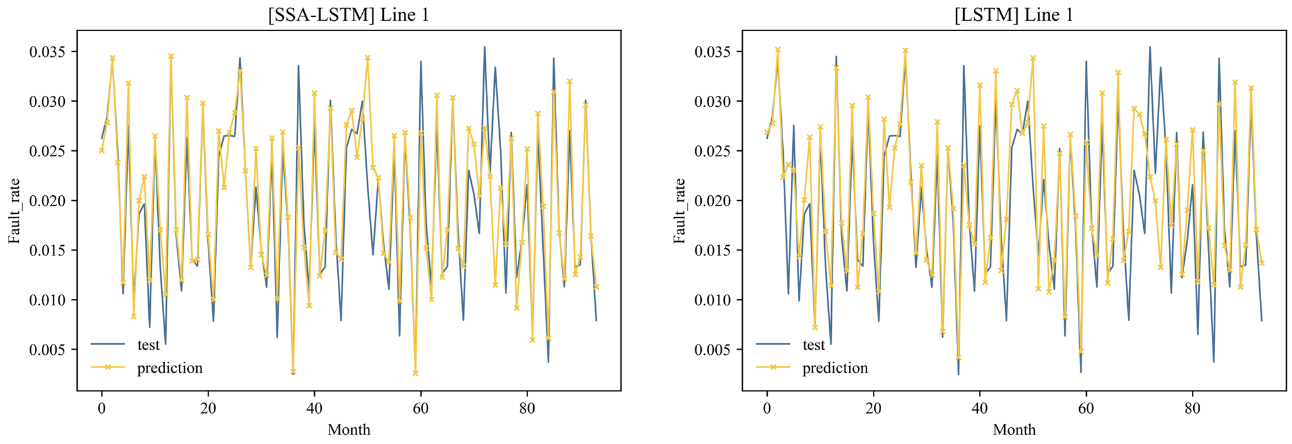
(**c**) (**d**)

Figure 12. (**a**) Prediction results for the PSD fault rate in the dataset containing all lines when using SSA-CNN-LSTM; (**b**) prediction results for the PSD fault rate in the dataset containing all lines when using CNN-LSTM; (**c**) prediction results for the PSD fault rate in the dataset containing all lines when using SSA-LSTM; (**d**) prediction results for the PSD fault rate in the dataset containing all lines when using LSTM.



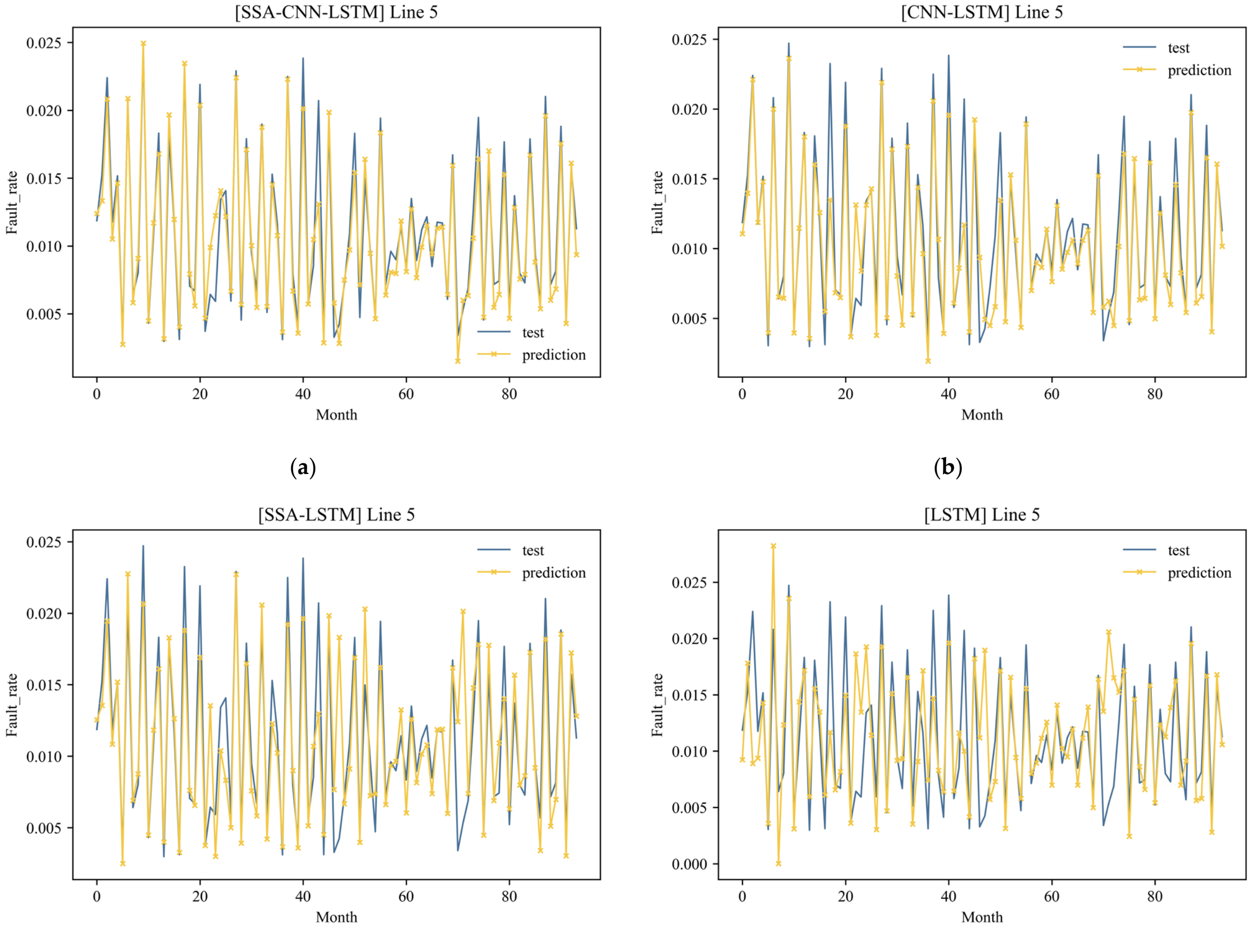
(**a**) (**b**)

Figure 13. *Cont*.



(**c**) (**d**)

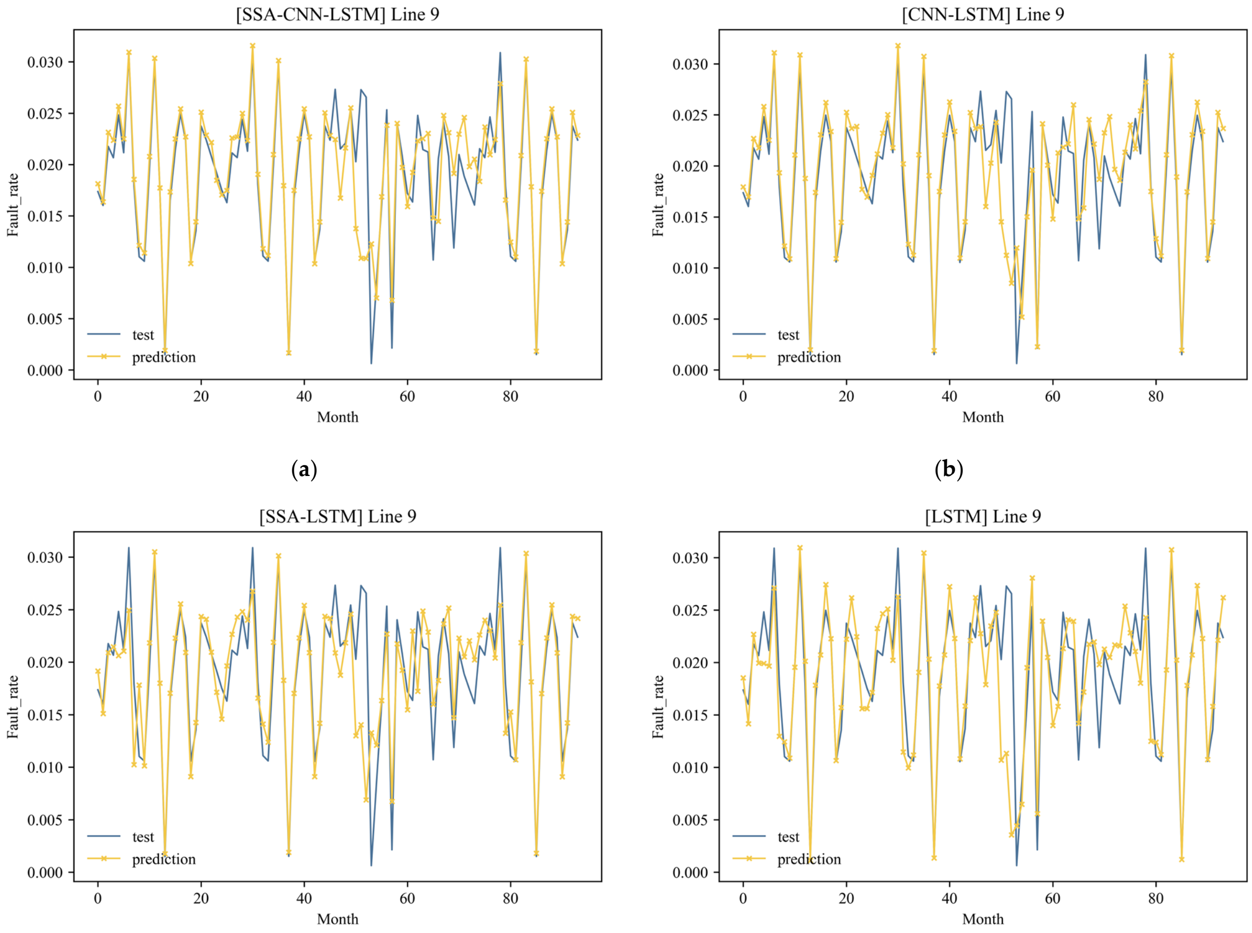
Figure 13. (**a**) Prediction results for the PSD fault rate in the Line 1 dataset when using SSA-CNN-LSTM; (**b**) prediction results for the PSD fault rate in the Line 1 dataset when using CNN-LSTM; (**c**) prediction results for the PSD fault rate in the Line 1 dataset when using SSA-LSTM; (**d**) prediction results for the PSD fault rate in the Line 1 dataset when using LSTM.



(**a**) (**b**)

(**c**) (**d**)

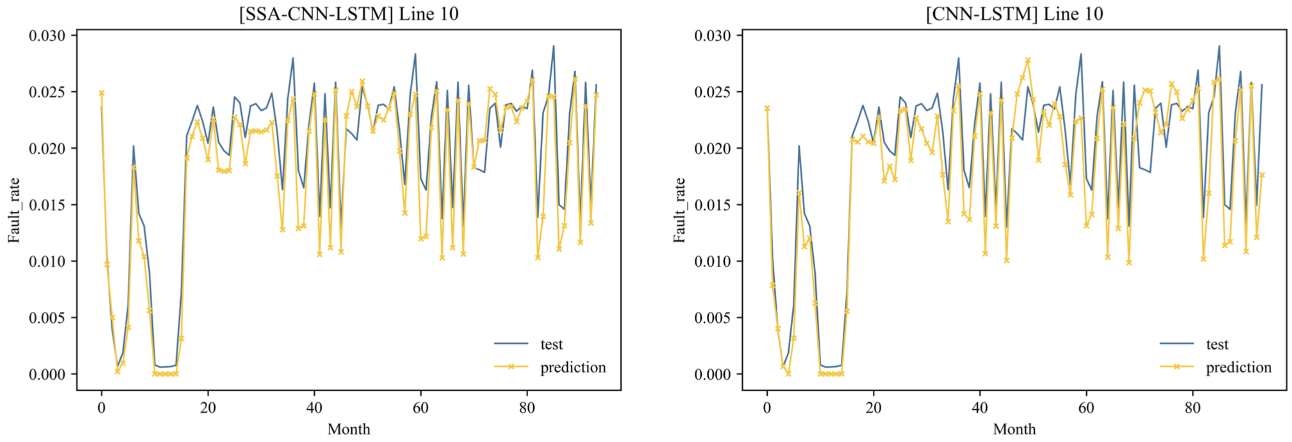
Figure 14. (**a**) Prediction results for the PSD fault rate in the Line 5 dataset when using SSA-CNN-LSTM; (**b**) prediction results for the PSD fault rate in the Line 5 dataset when using CNN-LSTM; (**c**) prediction results for the PSD fault rate in the Line 5 dataset when using SSA-LSTM; (**d**) prediction results for the PSD fault rate in the Line 5 dataset when using LSTM.



(**a**) (**b**)

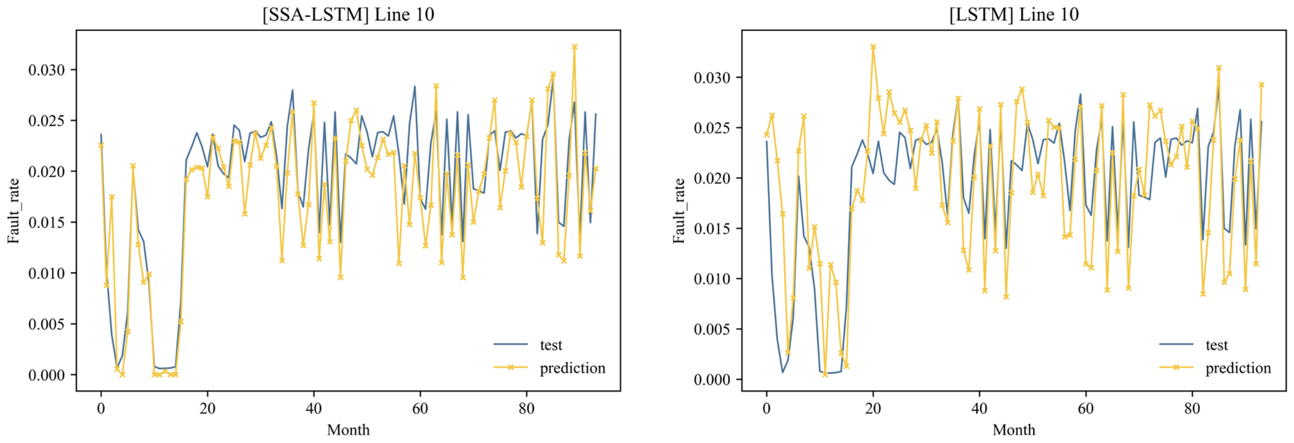
(**c**) (**d**)

Figure 15. (**a**) Prediction results for the PSD fault rate in the Line 9 dataset when using SSA-CNN-LSTM; (**b**) prediction results for the PSD fault rate in the Line 9 dataset when using CNN-LSTM; (**c**) prediction results for the PSD fault rate in the Line 9 dataset when using SSA-LSTM; (**d**) prediction results for the PSD fault rate in the Line 9 dataset when using LSTM.



(**a**) (**b**)

Figure 16. *Cont.*



(**c**) (**d**)

Figure 16. (**a**) Prediction results for the PSD fault rate in the Line 10 dataset when using SSA-CNN-LSTM; (**b**) prediction results for the PSD fault rate in the Line 10 dataset when using CNN-LSTM; (**c**) prediction results for the PSD fault rate in the Line 10 dataset when using SSA-LSTM; (**d**) prediction results for the PSD fault rate in the Line 10 dataset when using LSTM.

5. Conclusions

In this study, a prediction algorithm based on SSA-CNN-LSTM was proposed to predict the fault rate of platform screen door systems. Based on TimeGAN, augmented data that conformed to the temporal characteristics of the original dataset were generated, which solved the problem of sparse data in the original rail transit platform screen door fault dataset. The established SSA-CNN-LSTM fault rate prediction model extracted effective data features from low-dimensional data with insufficient feature depth through the convolutional layer, pooling layer, and other structures of a CNN, determined the optimal hyperparameters, automatically optimized the model network scale, solved the problem of the scale of the neural network model being difficult to determine, and realized the accurate prediction of the fault rate of rail transit platform screen door systems.

The results of this study show the following:

(1) The fault rate prediction model for rail transit platform screen doors based on SSA-CNN-LSTM was able to better fit nonlinear time-series data, such as the fault rate data of platform screen doors. Compared with the GRU model, the model performed better in prediction performance.

(2) The TimeGAN algorithm was used for data enhancement, the spatial and temporal distribution patterns of the data were effectively retained, and the data capacity was increased. TimeGAN learned the temporal characteristics of the fault rate data of platform screen door systems through the joint training of an automatic coding network and adversarial network, and it effectively solved the problem of data sparseness.

(3) Compared with the PSO-optimized CNN-LSTM model, when the SSA-optimized CNN-LSTM model was used to predict the PSD fault rate of all lines after enhancement, the RMSE was reduced by 0.001191 (improved by 45.2%), the MAE was reduced by 0.0007 (improved by 47.3%), and the R2 index was increased by 0.170016 (improved by 22.5%). When predicting the PSD fault rate of each line after enhancement, the performance of the CNN-LSTM model optimized with the SSA algorithm was better than that of the CNN-LSTM model optimized with the PSO algorithm. For example, in the fault rate prediction for Line 1, the RMSE was reduced by 0.00071 (improved by 22.2%), the MAE was reduced by 0.00072 (improved by 28.9%), and in the fault rate prediction for Line 10, R2 was increased by 0.0613 (improved by 7.6%).

In the future, further optimization of the SSA will be considered so that the results of multiple intelligent optimization parameters are close to the same. In addition, maintenance decisions for platform screen door systems based on fault rate prediction data will be considered.

Author Contributions: Conceptualization, J.X. and G.Y.; methodology, G.Y. and J.X.; formal analysis, Y.S. and J.X.; investigation, J.S. and Y.W.; resources, Y.W.; data curation, Y.W. and J.S.; verification, J.S. and J.X.; writing—original draft preparation, J.S., G.Y. and J.X.; writing—review and editing, Y.S., G.Y. and J.S.; visualization, G.Y. and J.S.; supervision, J.X.; project administration, Y.W. and J.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Ministry of Industry and Information Technology of China (No. 2022-A04R-1-1) and the Science and Technology Program of Shanghai, China (No. 22511104300).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data are confidential and were obtained from Shanghai Shentong Metro Group Co., Ltd.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Su, Z.; Li, X. Energy benchmarking analysis of subway station with platform screen door system in China. *Tunn. Undergr. Space Technol.* **2022**, *128*, 104655. [CrossRef]
2. Zhou, Q.; Lin, S.; Zhang, W. Analysis and Mitigation of Stray Current in Modern Metro Systems with OVPD and PSD. *IEEE Trans. Transp. Eelectr.* **2024**, *10*, 3153–3166. [CrossRef]
3. Lu, B.; Tong, D.; Chen, Q.; Shi, H.; Qi, R.; Bao, Y. Research and Design on Control System of Subway Platform Screen Door. *J. Shanghai Univ. Eng. Sci.* **2018**, *32*, 138–140+156.
4. Khalouli, S.; Benmansour, R.; Hanafi, S. An ant colony algorithm based on opportunities for scheduling the preventive railway maintenance. In Proceedings of the 2016 International Conference on Control, Decision and Information Technologies (CoDIT), St Pauls Bay, Malta, 6–8 April 2016; pp. 594–599.
5. Argyropoulou, K.; Iliopoulou, C.; Kepaptsoglou, K. Model for corrective maintenance scheduling of rail transit networks: Application to athens metro. *J. Infrastruct. Syst.* **2019**, *25*, 04018035. [CrossRef]
6. Gong, Q.; Yang, L.; Li, Y.H.; Xue, B. Dynamic Preventive Maintenance Optimization of Subway Vehicle Traction System Considering Stages. *Appl. Sci.* **2022**, *12*, 8617. [CrossRef]
7. Wu, S.M.; Castro, I.T. Maintenance policy for a system with a weighted linear combination of degradation processes. *Eur. J. Oper. Res.* **2020**, *280*, 124–133. [CrossRef]
8. *BS EN 13 306:2010*; Maintenance-Maintenance Terminology. Standards Policy and Strategy Committee. The British Standards Institution: Milton Keynes, UK, 2010.
9. Yilboga, H.; Eker, Ö.F.; Güçlü, A.; Camci, F. Failure prediction on railway turnouts using time delay neural networks. In Proceedings of the 2010 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications, Taranto, Italy, 6–8 September 2010; pp. 31–37.
10. Wang, Q.; Bu, S.Q.; He, Z.Y. Achieving predictive and proactive maintenance for high-speed railway power equipment with LSTM-RNN. *IEEE T. Ind. Inform.* **2020**, *16*, 650917. [CrossRef]
11. Wang, X.Z.; Xu, P.P.; Yang, Q.J.; Wu, G.L.; Wei, F. Fault Prediction Method of Access Control Terminal Based on Euclidean Distance Center SMOTE Method. In Proceedings of the 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS), Nanjing, China, 23–25 November 2018; pp. 84–89.
12. Trinh, H.C.; Kwon, Y.K. A Data-Independent Genetic Algorithm Framework for Fault-Type Classification and Remaining Useful Life Prediction. *Appl. Sci.* **2020**, *10*, 368. [CrossRef]
13. Duan, F.; Zhang, S.; Yan, Y.Z.; Cai, Z.Q. An Oversampling Method of Unbalanced Data for Mechanical Fault Diagnosis Based on MeanRadius-SMOTE. *Sensors* **2022**, *22*, 5166. [CrossRef]
14. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
15. Fernandez, A.; Garcia, S.; Herrera, F.; Chawla, N.V. SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *J. Artif. Intell. Res.* **2018**, *61*, 863–905. [CrossRef]
16. Li, Z.; Ma, C.; Shi, X.C.; Zhang, D.A.; Li, W.; Wu, L.B. Tsa-gan: A robust generative adversarial networks for time series augmentation. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Beijing, China, 18–22 July 2021; pp. 1–8.

17. Yoon, J.; Jarrett, D.; Van der Schaar, M. Time-series generative adversarial networks. In Proceedings of the 33rd Conference on Neural Information Proceeding Systems (NeurlPS), Vancouver, BC, Canada, 8–14 December 2019; pp. 23–50.
18. Zhu, H.H.; Wang, X.; Chen, X.Q.; Zhang, L.Y. Similarity search and performance prediction of shield tunnels in operation through time series data mining. *Automat. Constr.* **2020**, *114*, 103178. [CrossRef]
19. Lim, B.; Zohren, S. Time-series forecasting with deep learning: A survey. *Philos. Trans. R. Soc. A* **2021**, *379*, 20200209. [CrossRef] [PubMed]
20. Guo, J.W.; Lao, Z.P.; Hou, M.; Li, C.; Zhang, S.H. Mechanical fault time series prediction by using EFMSAE-LSTM neural network. *Measurement* **2021**, *173*, 108566. [CrossRef]
21. Guo, L.; Li, R.; Jiang, B. A data-driven long time-series electrical line trip fault prediction method using an improved stacked-informer network. *Sensors* **2021**, *21*, 4466. [CrossRef]
22. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 27–30 June 2016; pp. 770–778.
23. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *Comput. Sci.* **2014**, *140*, 91556.
24. Huang, G.; Liu, Z.; Van der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
25. Zha, W.T.; Jin, Y.; Li, Z.Y.; Li, Y.L. Wind Power Modeling based on Data Augmentation and Stacking Integrated Learning. In Proceedings of the 41st Chinese Control Conference (CCC), Hefei, China, 25–27 July 2022; pp. 5554–5559.
26. Sun, S.Y.; Zhang, G.; Liang, W.G.; She, B.; Tian, F.Q. Remaining useful life prediction method of rolling bearing based on time series data augmentation and BLSTM. *Syst. Eng. Electron.* **2022**, *44*, 10608.
27. Sun, Y.B.; Gui, W.H.; Chen, X.F.; Xie, Y.F. Evaluation model of aluminum electrolysis cell condition based on multi-source heterogeneous data fusion. *Int. J. Mach. Learn. CYB* **2023**, *23*, 1375–1396. [CrossRef]
28. Fan, S.K.S.; Tsai, D.M.; Yeh, P.C. Effective Variational-Autoencoder-Based Generative Models for Highly Imbalanced Fault Detection Data in Semiconductor Manufacturing. *IEEE. Trans. Semiconduct. Manuf.* **2023**, *36*, 205–214. [CrossRef]
29. Shi, J.; Ding, Y.; Lv, Z. An intermittent fault data generation method based on lstm and gan. In Proceedings of the 2021 Global Reliability and Prognostics and Health Management (PHM-Nanjing), Nanjing, China, 15–17 October 2021; pp. 2–12.
30. Sabir, R.; Rosato, D.; Hartmann, S.; Guehmann, C. Signal generation using 1d deep convolutional generative adversarial networks for fault diagnosis of electrical machines. In Proceedings of the 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 3907–3914.
31. Jain, A.; Kumar, A.M. Hybrid neural network models for hydrologic time series forecasting. *Appl. Soft Comput.* **2007**, *7*, 585–592. [CrossRef]
32. Moldovan, A.M.; Buzdugan, M.I. Prediction of Faults Location and Type in Electrical Cables Using Artificial Neural Network. *Sustainability* **2023**, *15*, 6162. [CrossRef]
33. Yang, C.L.; Yang, C.Y.; Chen, Z.X.; Lo, N.W. Multivariate time series data transformation for convolutional neural network. In Proceedings of the 2019 IEEE/SICE International Symposium on System Integration (SII), Paris, France, 14–16 January 2019; pp. 188–192.
34. Kashiparekh, K.; Narwariya, J.; Malhotra, P.; Vig, L.; Shroff, G. Convtimenet: A pre-trained deep convolutional neural network for time series classification. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–8.
35. Wu, Y.C.; Zhao, R.Z.; Ma, H.R.; He, Q.; Du, S.H.; Wu, J. Adversarial domain adaptation convolutional neural network for intelligent recognition of bearing faults. *Measurement* **2022**, *195*, 230–253. [CrossRef]
36. Che, Z.; Purushotham, S.; Cho, K.; Sontag, D.; Liu, Y. Recurrent neural networks for multivariate time series with missing values. *Sci. Rep.* **2018**, *8*, 6085. [CrossRef] [PubMed]
37. Bezyan, B.; Zmeureanu, R. Detection and Diagnosis of Dependent Faults That Trigger False Symptoms of Heating and Mechanical Ventilation Systems Using Combined Machine Learning and Rule-Based Techniques. *Energies* **2022**, *15*, 1691. [CrossRef]
38. Kan, A.; Zhang, Z.; Saligrama, V. Rnns incrementally evolving on an equilibrium manifold: A panacea for vanishing and exploding gradients? In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 5–9 May 2019; pp. 2360–2368.
39. Hocheriter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
40. Xu, H.L.; Ma, R.Z.; Yan, L.; Ma, Z.M. Two-stage prediction of machinery fault trend based on deep learning for time series analysis. *Digit. Signal Process.* **2021**, *117*, 1239. [CrossRef]
41. Zhang, C.; Chen, P.; Jiang, F.; Xie, J.; Yu, T. Fault Diagnosis of Nuclear Power Plant Based on Sparrow Search Algorithm Optimized CNN-LSTM Neural Network. *Energies* **2023**, *16*, 2934. [CrossRef]
42. Liu, K.Z.; Gou, J.Q.; Luo, Z.; Wang, K.; Xu, X.W.; Zhao, Y.J. Prediction of Dissolved Gas Concentration in Transformer Oil Based on PSO-LSTM Model. *Power Syst. Technol.* **2020**, *44*, 2778–2784.
43. Zhang, Y.S. Research on Key Technologies of Remaining Useful Life Estimation for Industrial Equipment Based on Deep Learning. Master's Thesis, University of Electronic Science and Technology of China, Chengdu, China, 2022.
44. Abdeljaber, O.; Avci, O.; Kiranyaz, S.; Gabbouj, M.; Inman, D.J. Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *J. Sound Vib.* **2017**, *388*, 154–170. [CrossRef]

45. Xue, J.; Shen, B. A novel swarm intelligence optimization approach: Sparrow search algorithm. *Syst. Sci. Control Eng.* **2020**, *8*, 22–34. [CrossRef]
46. Li, T.; Hua, M.; Wu, X. A hybrid cnn-lstm model for forecasting particulate matter (pm2.5). *IEEE Access* **2020**, *8*, 26933–26940. [CrossRef]
47. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; pp. 1942–1948.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.