

## Research Article

# A Prediction Method for the RUL of Equipment for Missing Data

**Chen Wenbai** <sup>1</sup>, **Liu Chang** <sup>2</sup>, **Chen Weizhao**<sup>1</sup>, **Liu Huixiang**<sup>1</sup>, **Chen Qili** <sup>1</sup>,  
and **Wu Peiliang** <sup>3</sup>

<sup>1</sup>School of Automation, Beijing Information Science & Technology University, Beijing 100101, China

<sup>2</sup>National Engineering Research Center for Information Technology in Agriculture, Beijing 100097, China

<sup>3</sup>School of Information Science & Technology, Yanshan University, Qinhuangdao 066004, China

Correspondence should be addressed to Chen Wenbai; chenwb03@126.com and Liu Chang; changliu22@outlook.com

Received 30 August 2021; Accepted 3 November 2021; Published 20 December 2021

Academic Editor: Long Wang

Copyright © 2021 Chen Wenbai et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present a prediction framework to estimate the remaining useful life (RUL) of equipment based on the generative adversarial imputation net (GAIN) and multiscale deep convolutional neural network and long short-term memory (MSDCNN-LSTM). The method we proposed addresses the problem of missing data caused by sensor failures in engineering applications. First, a binary matrix is used to adjust the proportion of “0” to simulate the number of missing data in the engineering environment. Then, the GAIN model is used to impute the missing data and approximate the true sample distribution. Finally, the MSDCNN-LSTM model is used for RUL prediction. Experiments are carried out on the commercial modular aero-propulsion system simulation (C-MAPSS) dataset to validate the proposed method. The prediction results show that the proposed method outperforms other methods when packet loss occurs, showing significant improvements in the root mean square error (RMSE) and the score function value.

## 1. Introduction

Prognosis and Health Management (PHM) aims to monitor, predict, and manage the health of the system through models and algorithms and is widely used in aviation, military equipment, industrial manufacturing, and other fields [1]. As one of the important research issues of PHM, remaining useful life prediction (RUL) can provide strategy support for establishing the best maintenance management for equipment. The data-driven method for RUL prediction is developed to analyze the equipment operation data through modeling to determine the remaining available time of equipment. Therefore, the quality of the data is directly related to the accuracy of the RUL prediction [2].

Precision equipment and multisensor fusion are widely used in the industrial field, and obtaining complete monitoring data is crucial to predicting the remaining useful life (RUL) of equipment. In engineering applications, various factors, such as failure of data storage, sensor damage, and mechanical failure, may lead to missing information during

equipment information collection and storage [3]. Data packet loss is especially detrimental in complex and harsh working environments, such as aerospace and agricultural production environments [4]. The high cost and difficulty of obtaining equipment degradation data and the existence of information intervals between samples make RUL prediction challenging.

In 1987, Rubin [5] proposed that missing data mechanisms fall into three categories: missing at random (MAR), missing completely at random (MCAR), and missing not at random (MNAR). Handling missing data appropriately is particularly important to ensure the accuracy of missing data imputation [6]. Scholars have conducted numerous studies, and the methods can be roughly divided into three categories: ignoring data or deletion, imputation, and statistical models.

Deleting the missing items in the dataset is the simplest data processing method. Strike et al. [7] simulated three types of mechanisms for dealing with missing data and used different techniques for processing missing data, such as

listwise deletion, mean imputation, and eight types of hot-deck imputation. A detailed simulation study was carried out, and it was concluded that simple deletion was a suitable choice when the missing data volume was small. The imputation method fills in missing values. The most probable value is typically used for imputation, which causes less information loss than incomplete samples obtained by deleting all missing values. Commonly used methods include mean imputation, median imputation, mode imputation, and maximum likelihood estimation. Inspired by machine learning, prediction models were used to estimate missing values from the available information in the dataset [8]. Troyanskaya et al. [9] used the k-nearest neighbor (KNN) to estimate missing values in gene microarray data. The imputation effect was better than the imputation method based on singular value decomposition (SVD). Duan et al. [10] used a deep learning model with a denoising stacked autoencoder (DSAE) to estimate missing values in traffic data. This method proved effective for traffic data imputation and analysis. A statistical model was used to impute the missing values based on the linear or nonlinear relationship between the missing data and the observed data. Ni et al. [11] proposed an advanced calculation method based on a Bayesian network to learn from the raw data. A Markov chain Monte Carlo method was used for sampling based on the probability distribution learned by the Bayesian network. It imputes the missing data multiple times and makes statistical inferences about the results. Li et al. [12] proposed a systematic calculation method of traffic flow data based on probabilistic principal component analysis and historical data to estimate missing flow data. A statistical model was used to impute the missing values based on the prior knowledge of the data model, providing excellent results. However, the statistical model has shortcomings due to the incomplete dataset and incomplete prior knowledge. Machine learning has substantial application potential for data imputation. This study focuses on exploiting the use of existing data and machine learning algorithms to impute missing values.

We propose an RUL prediction framework based on data imputation to deal with missing sensor data in engineering applications. First, the missing data are simulated using various missing sample rates. Then, the generative adversarial imputation net (GAIN) model is used to impute the missing values and fill in the dataset. Finally, the proposed multiscale deep convolutional neural network and long short-term memory (MSDCNN-LSTM) prediction model is used to obtain the RUL value of the equipment. The proposed method is well suited for predicting the RUL of equipment if the sensor data are affected by data packet loss in engineering applications. The performance of the proposed method is demonstrated using the commercial modular aero-propulsion system simulation (C-MAPSS) dataset.

## 2. Related Work

In recent years, deep learning has powerful function mapping capabilities and data processing capabilities. To extract the complex characteristics inside the spectrum and predict

the nicotine volume in tobacco, Jiang et al. [13] proposed a one-dimensional fully convolutional network (1D-FCN) model. Hu et al. [14] presented a deep neural network-based visual analysis approach to process videos to detect different augmentative and alternative communication users in practice sessions.

Deep learning has also been widely used in data-driven RUL prediction methods. Babu et al. [15] first tried to use convolutional neural network (CNN) to predict the RUL of the engine, which improved the ability to automatically extract multidimensional features. Then, Li et al. [16] improved the prediction accuracy by using the deep CNN (DCNN) structure and time window data processing. In order to make the CNN model learn more detailed features, Li et al. [17] proposed an algorithm of MSDCNN, that is, the DCNN with different convolution kernel sizes. In order to extract the time correlation features of condition monitoring data, Kong et al. [17] proposed a hybrid algorithm of CNN and long short-term memory (LSTM) to learn spatial and temporal features. Huang et al. [18] developed a novel deep convolutional neural network-bootstrap-based integrated prognostic approach for the remaining useful life (RUL) prediction of rolling bearing. Hu et al. [19] applied the long short-term memory (LSTM) model for RUL prediction of turbine engines and studied a parameter optimization method with Bayesian theory.

In this article, we use the RUL prediction model of MSDCNN-LSTM proposed by Liu et al. [20] to learn more detailed features in a high-dimensional space and predict RUL of aircraft engines. The hybrid MSDCNN-LSTM model consists of an MSDCNN submodel and an LSTM submodel. Among the MSDCNN-LSTM model, the MSDCNN is used to extract high-dimensional features from the input data by time window processing, and the LSTM performs time-series learning on the input data at the same time. Then, the feature map of MSDCNN and LSTM are added and flatten. Finally, the output is sent to a dense layer that represents the RUL output value. The structure chart of MSCNN-LSTM is shown in Figure 1.

## 3. Proposed Method

*3.1. Missing Data Imputation Method Based on GAIN.* In 2014, Lan Goodfellow et al. [21] first proposed the generative adversarial net (GAN), which generates data in an adversarial manner with generators and discriminators. The method attracted the attention of researchers and was verified theoretically and practical in engineering applications. The GAN has wide applicability in image, text, and audio processing and other fields.

We used the GAIN model [22] to generate time-series data with a similar distribution as the original for missing data imputation. The basic structure of the model is shown in Figure 2.

The generator is used to observe each part of the real data, and the missing data are imputed according to the observations. The vector  $\bar{X}$  in the missing data imputation is expressed as follows:

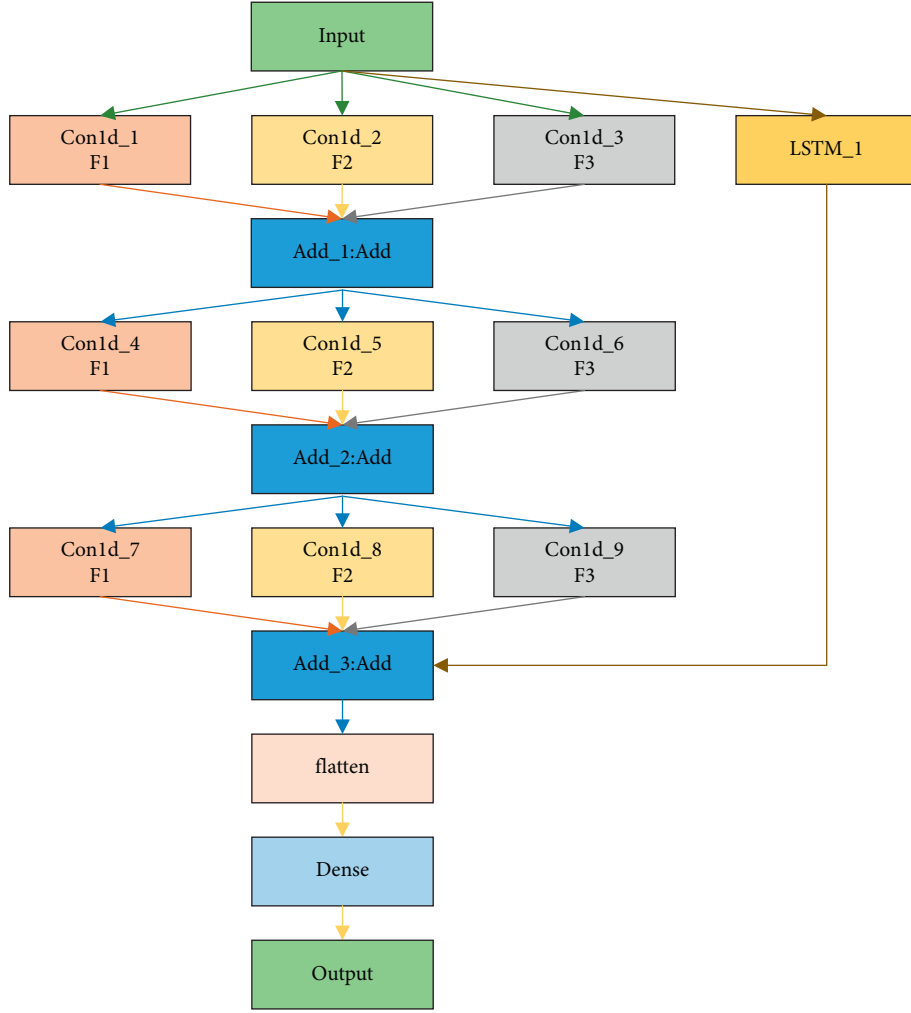


FIGURE 1: The structure chart of MSCNN-LSTM.

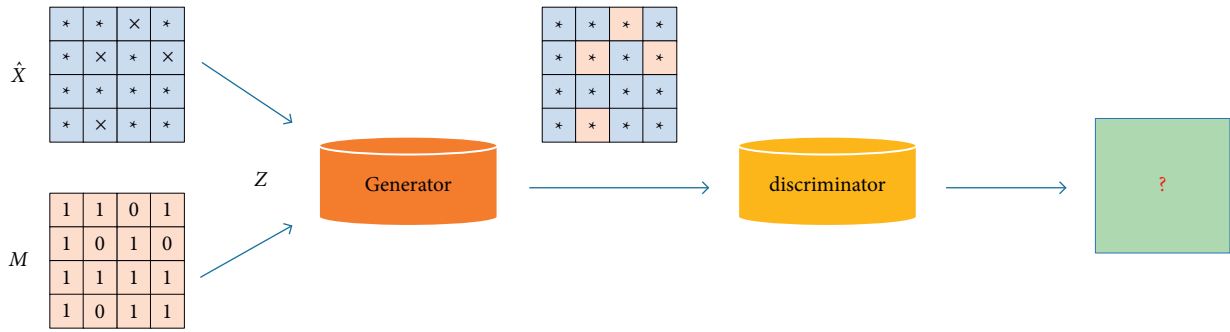


FIGURE 2: The basic structure of GAIN.

$$\bar{X} = G(\hat{X}, M, (1 - M) \odot Z), \quad (1)$$

$$\tilde{X} = M \odot \hat{X} + (1 - M)\bar{X}. \quad (2)$$

where  $\hat{X}$  represents a small sample with missing data,  $M$  represents a binary matrix with the same size as  $\hat{X}$ ,  $Z$  represents noise, and  $\odot$  represents the multiplication of the corresponding elements.

Finally, the generator outputs a complete vector  $\tilde{X}$  after imputation as follows:

Since some of the output of the generator is real and some is generated, the difference between the GAIN and the GAN is that the discriminator of GAIN does not determine the authenticity of the entire vector but detects the real and generated parts, i.e., it predicts the value of  $m$  in  $M$ . The model trains  $D$  by maximizing the probability of correctly

predicting  $M$  and trains  $G$  by minimizing the probability of correctly predicting  $M$ . The objective function is expressed as

$$\min_G \max_D V(G, D) = E_{\tilde{X}, M} [M^T \log D(\tilde{X}) + (1 - M)^T \log(1 - D(\tilde{X}))]. \quad (3)$$

The discriminator distinguishes the source of each part of the input data, and the obtained discriminant matrix is represented by  $\hat{M}$ . The cross-entropy loss function is used to evaluate each element in  $M$ :

$$\mathcal{L}_D(m, \hat{m}) = \left[ \sum_{i=1}^d m_i \log(\hat{m}_i) + (1 - m_i) \log(1 - \hat{m}_i) \right]. \quad (4)$$

The loss function of the generator is defined as

$$\mathcal{L}_G(m, \hat{m}) = \left[ \sum_{i=1}^d 1 - m_i \log(\hat{m}_i) \right] + \alpha \text{MSE}. \quad (5)$$

**3.2. RUL Prediction Framework Based on GAIN and MSDCNN-LSTM.** An RUL prediction framework that combines the GAIN and MSDCNN-LSTM is designed; it consists of three parts: preprocessing the missing data, missing data imputation based on the GAIN model, and RUL prediction based on the MSDCNN-LSTM model, as shown in Figure 3.

First, data preprocessing is performed on the C-MAPSS dataset. The method described in Section 1 is used to construct sample data for the training set with different missing data rates. Subsequently, the GAIN network is used to impute the samples with different missing data rates, and the generator and the discriminator generate data in an adversarial manner to obtain a dataset close to the original one. Finally, the generated samples are used as the input of the MSDCNN-LSTM prediction model. The MSDCNN and LSTM models process the data simultaneously. The multiscale structure of the MSDCNN substantially improves the feature extraction capability. Convolution kernels of different sizes ( $F1$ ,  $F2$ , and  $F3$ ) are used to extract features from the input data, and the feature maps are spliced together and combined with the time series to obtain the prediction results of the LSTM. Continuous iteration is used to evaluate the trained model using two indicators (root mean square error (RMSE) and the score function), and the test set data are input to obtain the RUL prediction result.

The RUL prediction process based on the combination of GAIN and MSDCNN-LSTM is shown in Figure 4. First, the missing data are generated using missing data rates of 0.1, 0.2, 0.3, 0.4, and 0.5. Then, we use the GAIN network to impute the missing values of the samples. During the training of the GAIN network, we set the epoch to 1000 times, and the newly generated time-series dataset are standardized and processed by a time window. After setting the RUL labels of the training set and test set, the next stage is model training and system prediction. The parallel MSDCNN-LSTM hybrid model performs multiscale feature

extraction on the time-series training set, and the parameters and weights in the model are updated using a minibatch of 512. When the early stopping conditions set by the system are met, the model training ends early. If the early stopping condition is not met, the minibatch training is continued until the maximum epoch. After the model is trained, we input the test set data to predict the RUL result of each engine.

## 4. Experimental and Results

**4.1. Experimental Dataset and Settings.** The experiments are conducted on a server computer configured with an Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10 GHz and an NVIDIA GeForce TITAN XP. The C-MAPSS dataset is used to verify the proposed method. The C-MAPSS dataset is divided into four subsets (FD001, FD002, FD003, and FD004) according to the operating conditions and failure modes. Each dataset contains engine degradation data monitored by 21 sensors, as listed in Table 1. Each subset is divided into a training set and a test set. FD002 and FD004 have 6 operating conditions, and FD003 and FD004 have 2 failure modes.

The score function and the RMSE are used as evaluation indicators. The formula of the score function is [23]

$$\text{Score} = \begin{cases} \sum_{i=1}^N (e^{-d_i/13} - 1), & d_i < 0, \\ \sum_{i=1}^N (e^{d_i/10} - 1), & d_i \geq 0. \end{cases} \quad (6)$$

The formula of the RMSE is

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N d_i^2}, \quad (7)$$

where  $d_i$  represents the difference between the predicted value of RUL and the true value,  $d_i = \text{RUL}_{\text{pre}} - \text{RUL}_{\text{act}}$ ,  $i = 1, 2, \dots, N$ . When  $d_i$  is less than 0, the predicted value is less than the true value, and the result is referred to as an advanced prediction; otherwise, it is a lagging prediction.

The lower the value of the score function and RMSE, the better the predictive ability of the model. The RMSE is a symmetric function and provides the same result for an advanced prediction and lagging prediction. However, the score function is an asymmetric function and is more sensitive to lagging prediction. Because lagging prediction has more serious consequences, it results in stronger penalties than advanced prediction. Therefore, these indicators comprehensively measure the performance of the algorithm.

**4.2. Simulating the Missing Data Rate.** It is assumed that the original dataset is  $X$ ,  $X = [X_{i1}, X_{i2}, \dots, X_{ij}, \dots, X_{im}] \in R^{m \times n}$ ,  $i \in [1, n]$  and  $j \in [1, m]$ , where  $m$  is the number of sensors,  $n$  is the length of the time series, and  $x_{ij}$  is the measured value of the  $j$ th sensor corresponding to the  $i$ th period. Here, we define a binary matrix  $M$ ,

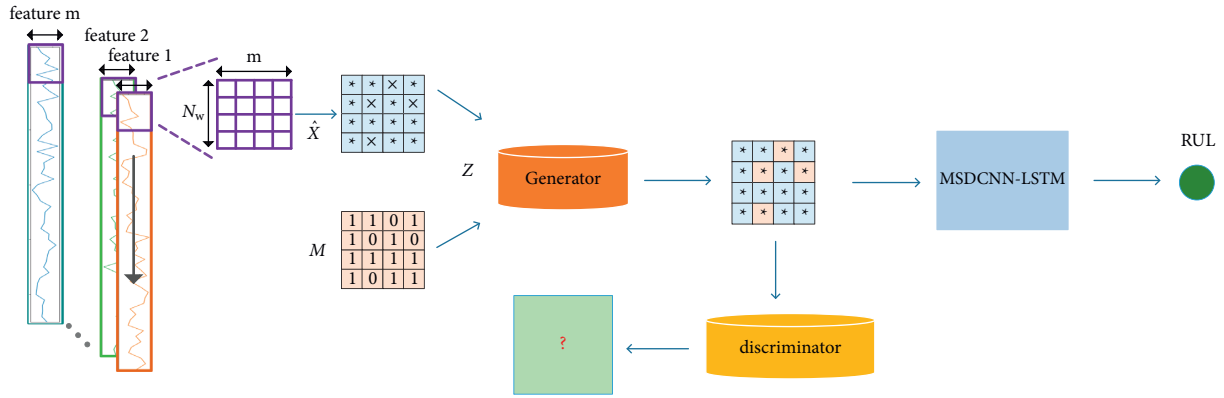


FIGURE 3: RUL prediction framework based on the GAIN and MSDCNN-LSTM.

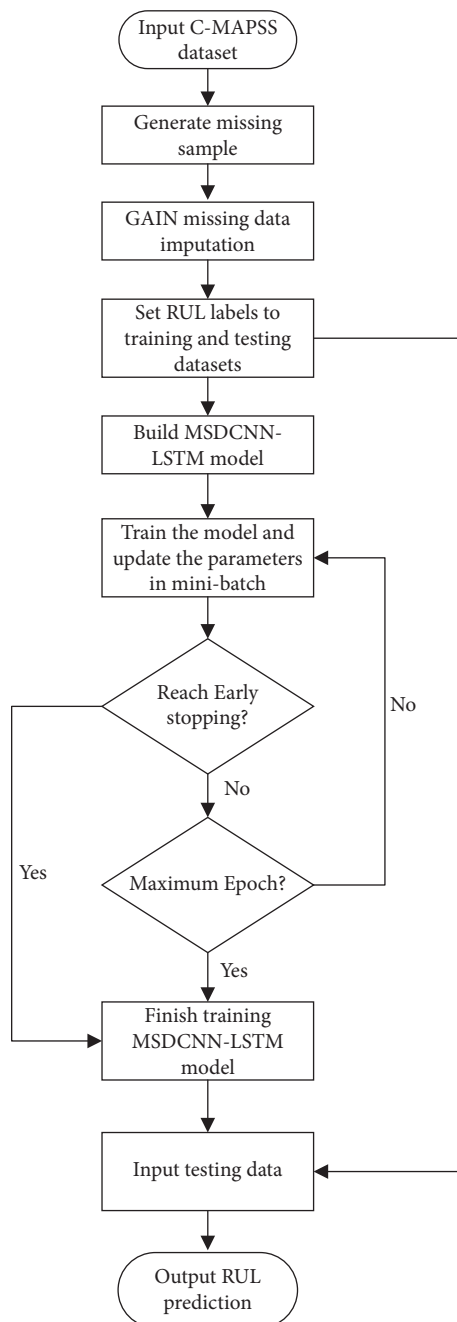


FIGURE 4: Flowchart of the RUL prediction.

TABLE 1: C-MAPSS subdatasets.

Subdatasets	FD001	FD002	FD003	FD004
Training engines	100	260	100	249
Test engines	100	259	100	248
Conditions	1	6	1	6
Failure modes	1	1	2	2

$M = [M_{i1}, M_{i2}, \dots, M_{ij}, \dots, M_{im}] \in R^{m \times n}$ ,  $i \in [1, n]$  and  $j \in [1, m]$ , which has the same size as the original data  $X$  and consists of 0 and 1 values. The reconstructed missing data  $\hat{X}$  can be expressed as

$$\hat{X} = \begin{cases} X_{i,j}, & M_{i,j} = 1, \\ \text{Nan}, & \text{other}, \end{cases} \quad (8)$$

where  $M$  represents the observed component of  $X$ . A value of 1 indicates the observed data, and a value of 0 represents the missing data  $\hat{X}$ . Datasets with different missing sample rates can be created by changing 0 to another value.

**4.3. Data Imputation Simulation Results and Analysis.** In the 4 subsets of the C-MAPSS dataset, 7 sensor data with no changes were eliminated. Therefore, the sensor numbers used in this experiment are 2, 3, 4, 7, 8, 9, 11, 12, 13, 14, 15, 17, 20, and 21. The RMSE was used as an evaluation indicator to evaluate the imputation effect of GAIN.

Table 2 lists the imputation accuracy of GAIN for different missing data rates (0.1, 0.2, 0.3, 0.4, and 0.5) on the C-MAPSS dataset.

As the missing data rate increases, the RMSE values of the four subdatasets FD001, FD002, FD003, and FD004 increase, and the accuracy decreases. In the case of a high missing data rate, there is less sample information, and it is difficult to fill in the missing sample data. The imputation performance of GAIN is better for the FD002 and FD004 datasets with complex working conditions and a large sample size than for the FD001 and FD003 datasets with simple working conditions and a small sample size. Therefore, the imputation performance is better for a larger sample size, and the prediction accuracy decreases as the missing data rate increases.

Figure 5 shows the visualization results of GAIN after missing data imputation for a missing data rate of 0.5. The horizontal axis represents the operating cycle of the first engine, and the vertical axis is the result of the first sensor data after maximum-minimum standardization. The middle black rectangle represents the real data, and the red dots represent the results of GAIN after imputation. Although the effect of missing data is more serious when the missing data rate is high, the data after imputation based on GAIN fluctuates in a small range around the real data, and the overall distribution is consistent with the real data distribution.

Table 3 shows the influence of the loss function on the GAIN model performance. During data imputation, the loss function is particularly important for training the generator and discriminator models. After conducting experiments, we found that the model performance was best when the

TABLE 2: The RMSE value of GAIN for different missing data rates.

Missing data rates	FD001	FD002	FD003	FD004
0.1	0.1041	0.0217	0.0915	0.0210
0.2	0.1050	0.0223	0.0917	0.0200
0.3	0.1076	0.0257	0.0988	0.0231
0.4	0.1079	0.0419	0.0997	0.0525
0.5	0.1091	0.1011	0.1126	0.0569

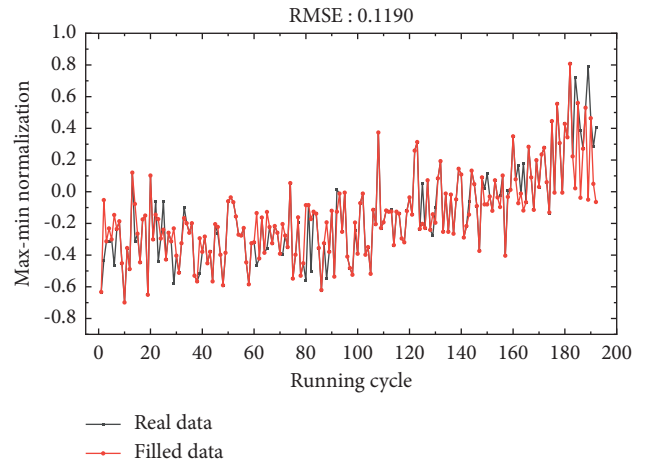


FIGURE 5: The result of GAIN imputation.

cross-entropy loss function and the mean square error loss function were used. We use the FD001 dataset with a missing data rate of 0.5 as an example to verify the impact of the loss function on the model performance and compare the simulation results obtained from different combinations of loss functions. It can be seen from Table 3 that the combinations of the two loss functions and the adjustment of the parameters  $\partial$  significantly affect the results. The optimal RMSE value is obtained when the cross-entropy loss function is used for the discriminator, and the cross-entropy loss function +  $\partial$  mean square are used for the generator. In the experiment, different combinations were used under the same conditions to verify the effect of the parameter  $\partial$  in the loss function. The results are listed in Table 4.

Adding the parameter  $\partial$  to the generator loss function improves the imputation accuracy, but  $1/\partial$  and  $(1 - \partial)$  substantially increase the RMSE value. Therefore, the model provides the best performance when the coefficient of the RMSE in the generator loss function is  $\partial$ .

Figure 6 shows the results of the GAIN imputation and other methods. The GAIN imputation, mean imputation, median imputation, and mode imputation are compared using the FD001 dataset. The horizontal axis represents the missing data rate, and the vertical axis represents the RMSE. As the missing data rate increases, the RMSEs of the four methods show an upward trend, and the imputation performance decreases. The results for different missing data rates indicate that the mean value imputation results are more stable than the mode and median imputation methods. However, the GAIN achieves the smallest RMSE values for the different missing data rates, indicating that it outperforms the other methods.

TABLE 3: The influence of the loss function on model performance.

D_loss	G_loss	RMSE
Cross entropy	Cross entropy + $\partial$ RMSE	0.1190
Cross entropy + $\partial$ RMSE	Cross entropy + $\partial$ RMSE	0.1347
Cross entropy	Cross entropy	0.5065
Cross entropy + $\partial$ RMSE	Cross entropy	0.4692
Cross entropy	RMSE	0.1423

TABLE 4: The influence of the parameters in the loss function on model performance.

D_loss	G_loss	RMSE
Cross entropy	Cross entropy + $\partial$ RMSE	0.1190
Cross entropy	Cross entropy + $\partial^2$ RMSE	0.1227
Cross entropy	Cross entropy + $1/\partial$ RMSE	0.4900
Cross entropy	Cross entropy + $(1 - \partial)$ RMSE	0.5664
Cross entropy	Cross entropy + RMSE	0.1245

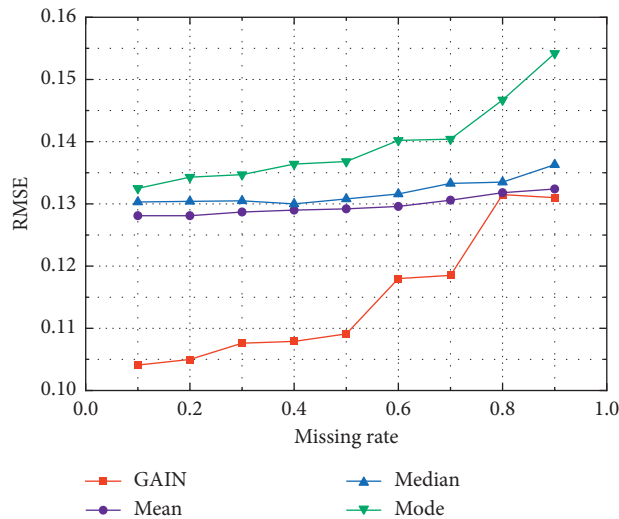


FIGURE 6: The results of different imputation methods for different missing data rates.

Figure 7 shows the RUL prediction results of all engine units after GAIN imputation on the C-MAPSS dataset when missing data rate is 0.1. The test engine is sorted by RUL from small to large to better observe the changes in prediction accuracy. The horizontal axis represents test engine unit, and the vertical axis represents the RUL. In the figure, the black dots represent the real RUL, and the red dots represent the model prediction results. It can be seen from Figure 7 that, at the initial stage of engine operation, the RUL value is relatively large and the prediction error is relatively large. When the engine runs for a long time or is about to fail, the degradation information is more obvious, and the predictive performance is significantly enhanced. The proposed framework reflects a good forecasting effect.

Table 5 shows the RUL prediction results with and without GAIN imputation for a missing data rate of 0.1. It is worth noting that the system automatically replaces missing data with 0 to ensure the smooth execution of the RUL prediction algorithm. Therefore, when the missing data rate

is 0.1, the score function value cannot be obtained, and it causes difficulties for the subsequent RUL prediction, such as a substantial increase in the RMSE value. However, after the missing data are imputed by GAIN, the prediction results are significantly improved. The RMSE has increased by at least 80.16%, and the score function value has increased by at least 99.98%.

Table 6 shows the prediction results of the proposed GAIN method for missing data rates of 0.1, 0.2, 0.3, 0.4, and 0.5. As the missing data rate increases, the prediction accuracy of the 4 subdatasets decreases. The score function is more affected than the RMSE. When the missing rate is less than 0.4, the proposed RUL prediction framework based on data imputation can show better performance. When the missing data rate is higher than or equal to 0.4, too much data information is lost, resulting in low prediction performance. However, the prediction result of the proposed framework is much better than using no missing data imputation for a missing data rate of 0.1.

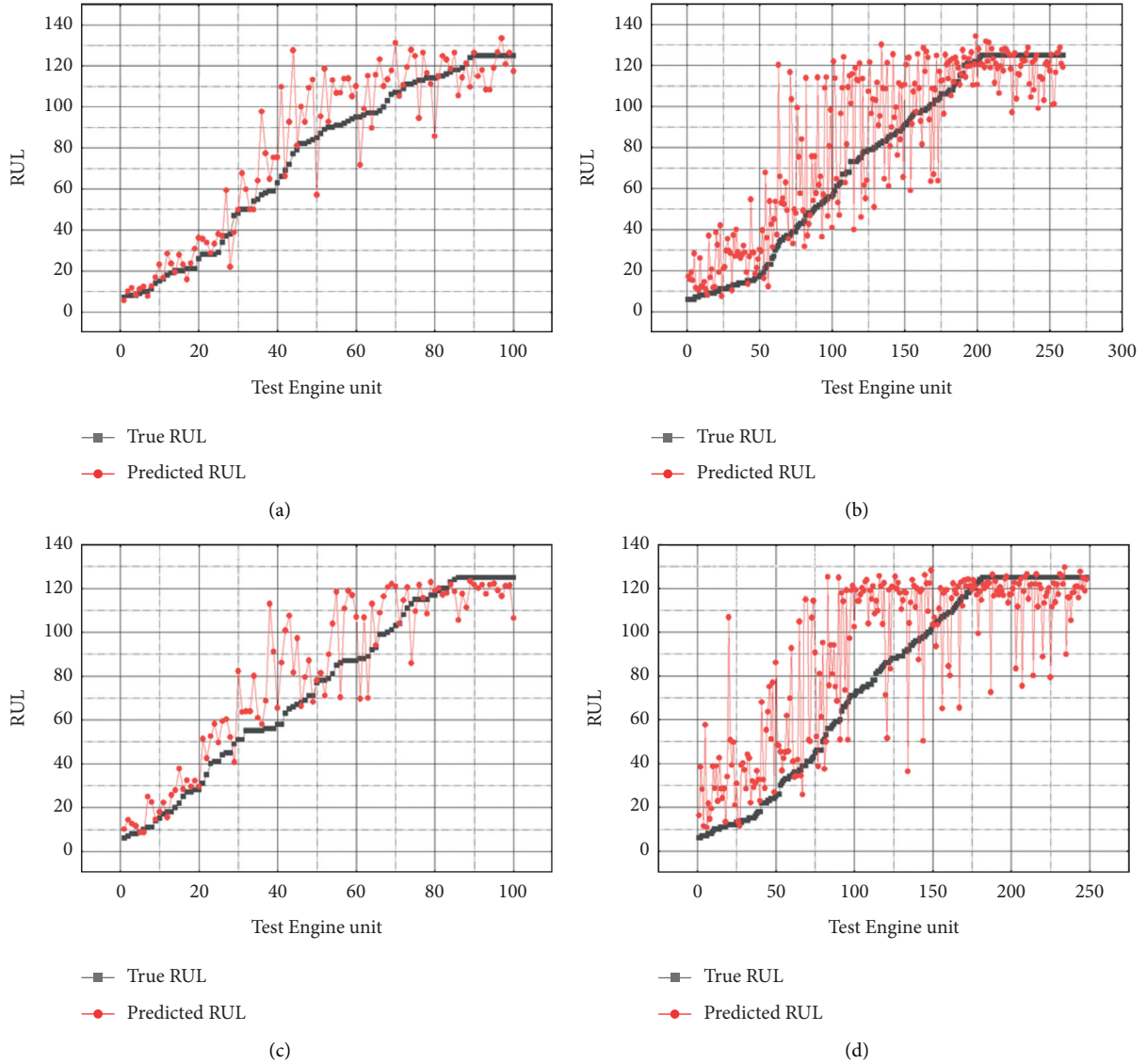


FIGURE 7: The results of RUL prediction after GAIN imputation when missing data rate is 0.1. (a) 100 test engine units on FD001. (b) 259 test engine units on FD002. (c) 100 test engine units on FD003. (d) 248 test engine units on FD004.

TABLE 5: The results of the RUL prediction with and without data imputation for a missing data rate of 0.1.

Method	FD001		FD002		FD003		FD004	
	RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score
GAIN imputation	14.48	274.62	18.73	2067.25	15.83	338.05	23.28	4750.18
No imputation	71.47	1620023.5	77.04	1815133.4	75.62	3656987.8	66.73	4859614.0

TABLE 6: The results of the RUL prediction with data imputation for different missing data rates.

Missing rate	FD001		FD002		FD003		FD004	
	RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score
0.1	14.48	274.62	18.73	2067.25	15.83	338.05	23.28	4750.18
0.2	15.74	361.07	23.62	3647.92	17.45	494.89	24.45	5744.81
0.3	18.17	506.06	23.74	3208.44	23.07	847.91	32.52	13376.09
0.4	19.61	521.18	43.19	36818.28	29.13	1531.91	32.07	42081.27
0.5	19.65	503.78	51.96	703764.25	26.44	1216.59	43.38	34483.86



## 5. Conclusions

This paper proposed a RUL prediction method based on the combination of GAIN and MSDCNN-LSTM. Experiments were carried out with a missing data rate of 0.1–0.5 to simulate data packet loss in industrial production. In the GAIN method, the generator interacts with the discriminator to impute the missing data and generate a sample dataset that is close to reality. This dataset is used as the input of the MSDCNN-LSTM prediction model. A comparison of the simulation results of GAIN and other methods indicated that the GAIN imputation method outperformed the mean, median, and mode imputation methods. The proposed prediction framework was compared with no data imputation when packet loss occurred and exhibited a significant improvement. The RUL prediction framework showed better prediction performance than other methods on the C-MAPSS dataset for different missing data rates.

## Data Availability

The data used to support the findings of the study are included within the article.

## Conflicts of Interest

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

## Acknowledgments

This work was supported by the Beijing Natural Science Foundation (Grant no. 4202026), Qin Xin Talents Cultivation Program of Beijing Information Science and Technology University (QXTCP A202102), Beijing Postdoctoral Science Foundation (Grant no. ZZ-2019-65), Chaoyang District Postdoctoral Science Foundation (Grant no. 2019ZZ-45), the National Key R&D Program of China (Grant no. 2018YFB1308300), and National Natural Science Foundation of China (Grant no. 62103056).

## References

- [1] B. Bagheri, M. Rezapoor, and J. Lee, "A unified data security framework for federated prognostics and health management in smart manufacturing," *Manufacturing Letters*, vol. 24, pp. 136–139, 2020.
- [2] S. F. L. T. M. Zhang and C. H. Hu, "Deep convolutional generative adversarial network based missing data generation method and its application in remaining useful life prediction," *Acta Aeronautica et Astronautica Sinica*, vol. 42, no. 6, (in Chinese), Article ID 625207, 2021.
- [3] X. Hu, H. Zhang, D. Ma, and R. Wang, "Hierarchical pressure data recovery for pipeline network via generative adversarial networks," *IEEE Transactions on Automation Science and Engineering*, pp. 1–11, 2021.
- [4] F. Qu, J. Liu, Y. Ma, D. Zang, and M. Fu, "A novel wind turbine data imputation method with multiple optimizations based on GANs," *Mechanical Systems and Signal Processing*, vol. 139, Article ID 106610, 2020.
- [5] D. B. Rubin, "The calculation of posterior distributions by data augmentation: comment: A n sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: the sir algorithm," *Journal of the American Statistical Association*, vol. 82, no. 398, pp. 543–546, 1987.
- [6] A. Nguetilbaye, H. Wang, D. A. Mahamat, and S. B. Junaidu, "Modulo 9 model-based learning for missing data imputation," *Applied Soft Computing*, vol. 103, Article ID 107167, 2021.
- [7] K. Strike, K. El Emam, and N. Madhavji, "Software cost estimation with incomplete data," *IEEE Transactions on Software Engineering*, vol. 27, no. 10, pp. 890–908, 2001.
- [8] J. M. Jerez, I. Molina, P. J. García-Laencina et al., "Missing data imputation using statistical and machine learning methods in a real breast cancer problem," *Artificial Intelligence in Medicine*, vol. 50, no. 2, pp. 105–115, 2010.
- [9] O. Troyanskaya, M. Cantor, G. Sherlock et al., "Missing value estimation methods for DNA microarrays," *Bioinformatics*, vol. 17, no. 6, pp. 520–525, 2001.
- [10] Y. Duan, Y. Lv, Y.-L. Liu, and F.-Y. Wang, "An efficient realization of deep learning for traffic data imputation," *Transportation Research Part C: Emerging Technologies*, vol. 72, pp. 168–181, 2016.
- [11] D. Ni and J. D. Leonard, "Markov chain Monte Carlo multiple imputation using bayesian networks for incomplete intelligent transportation systems data," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1935, no. 1, pp. 57–67, 2005.
- [12] L. Qu, L. Jianming, H. Li, and Y. Zhang, "PPCA-based missing data imputation for traffic flow volume: a systematic approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 512–522, 2009.
- [13] D. Jiang, G. Hu, G. Qi, and N. Mazur, "A fully convolutional neural network-based regression approach for effective chemical composition analysis using near-infrared spectroscopy in cloud," *Journal of Artificial Intelligence and Technology*, vol. 1, no. 1, pp. 74–82, 2021.
- [14] G. Hu, S. H. K. Chen, and N. Mazur, "Deep neural network-based speaker-aware information logging for augmentative and alternative communication," *Journal of Artificial Intelligence and Technology*, vol. 1, no. 2, pp. 138–143, 2021.
- [15] G. Sateesh Babu, P. Zhao, and X.-L. Li, "Deep convolutional neural network based regression approach for estimation of remaining useful life," in *International conference on database systems for advanced applications*, pp. 214–228, Springer, Dallas, TX, USA, April 2016.
- [16] X. Li, Q. Ding, and J.-Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Reliability Engineering & System Safety*, vol. 172, pp. 1–11, 2018.
- [17] H. Li, W. Zhao, Y. Zhang, and E. Zio, "Remaining useful life prediction using multi-scale deep convolutional neural network," *Applied Soft Computing*, vol. 89, Article ID 106113, 2020.
- [18] C. G. Huang, H. Z. Huang, Y. F. Li, and P. Weiwen, "A novel deep convolutional neural network-bootstrap integrated method for RUL prediction of rolling bearing," *Journal of Manufacturing Systems*, vol. 61, pp. 757–772, 2021.
- [19] S. Hu, S. Zhang, and X. Xu, "RUL prediction by LSTM model with bayesian parameter optimization for turbine engines," *Journal of Physics: Conference Series*, IOP Publishing, vol. 1646, no. 1, , Article ID 012122, 2020.

- [20] C. Liu and W. Chen, "A RUL prediction method of equipments based on MSDCNN-LSTM," *Xibei Gongye Daxue Xuebao/Journal of Northwestern Polytechnical University*, vol. 39, no. 2, pp. 407–413, 2021.
- [21] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza et al., *Generative Adversarial Networks*, <https://arxiv.org/abs/1406.2661>, 2014.
- [22] J. Yoon, J. Jordon, and M. Schaar, "Gain: missing data imputation using generative adversarial nets," in *International Conference on Machine Learning*, pp. 5689–5698, July 2018, <https://arxiv.org/abs/1806.02920>.
- [23] X. S. Si, W. Wang, C. H. Hu, and M. Y. Chen, "A Wiener-process-based degradation model with a recursive filter algorithm for remaining useful life estimation," *Mechanical Systems and Signal Processing*, vol. 35, no. 1-2, pp. 219–237, 2013.