Design and Application of Case Study on Missing Value Imputation in Industrial Time Series Data

Peiyuan Yang, Jin Hao

***Abstract*— With the rapid advancement of industrial informatization, the surge in data volume has become an undeniable fact. However, the problem of missing data is becoming increasingly prominent. Traditional missing value filling models are gradually becoming inadequate in terms of accuracy and precision, especially in complex industrial environments where these models often struggle to effectively cope with complex and ever-changing data processing needs. This article aims to explore in depth the data missing value filling method based on generative models, aiming to provide a more reliable and accurate solution for data integrity. This method encompasses the core theories, technical methods, and evaluation criteria of data processing and data generation modules, ensuring maximum accuracy and reliability of data during the data filling process. Taking the KDD2018cup dataset as the empirical research object, this study systematically analyzes the model design principles and application strategies for filling missing values in industrial data. Through in-depth analysis and careful design, we provide a new solution to the problem of missing data, which not only improves the filling accuracy but also enhances the data processing ability of the model in industrial environments. This research achievement will provide strong support for data management in the process of industrial informatization and promote further improvement of data quality.**

***Index Terms*—Missing data, Industrial time series data, Generate model, Generative adversarial network**

I. INTRODUCTION

With the solid progress of industrialization and the constant emergence of technological innovation, industrial production informatization has leapt to become the core driving force for industrial upgrading and enhancing core competitiveness[1]. However, in complex industrial environments, various factors such as equipment failures, sensor failures, communication link interruptions, and human operation omissions are intertwined, often leading to data loss. This challenge poses an undeniable threat to the coherence of industrial production and the accuracy of data analysis. Therefore, the processing and filling of these missing data has become a key link in ensuring stable production processes and reliable data quality[2].

Scholars have adopted high-performance generation models in the field of data interpolation, especially the GAN (generation countermeasure network) model proposed by Goodfellow[3] et al. in 2014. This model is famous for its antagonistic learning mechanism, which can accurately capture the distribution characteristics of the original data and generate new data that is highly consistent with the distribution of the original data set. Based on this principle, YOON[4] et al. further proposed a GAN interpolation method for missing data values. This method shields the missing position information and encourages the generator to create new samples that follow the distribution pattern of the original data. However, this method appears inadequate when dealing with temporal data, as it fails to fully consider the temporal dependence and continuity in temporal data. Subsequently, Chao[5] et al. developed a multi-modal missing data value interpolation technique based on GAN, which aims to learn common attributes between different modal data and effectively interpolate missing data for specific modalities based on this. However, multimodal data is not common in practical applications, which limits the widespread application of this method in practical scenarios. To make up for this deficiency, Luo[6] et al. proposed a missing value interpolation method for GAN temporal data based on recurrent neural networks. This innovative method performs well in processing temporal data and is one of the few techniques suitable for interpolating missing values in temporal data. However, this method also has some limitations. The generator network is built based on RNN (Recurrent Neural Network) and may encounter information forgetting issues when processing long sequence data. In addition, since the input of the generator is a random vector, after GAN training, additional time and computational resources are needed to find the optimal vector that matches each temporal data, which to some extent affects the time efficiency of the interpolation process. To address the aforementioned issues, this article provides an in-depth case study by combining the characteristics of missing values in industrial time series data. Through in-depth analysis of the characteristics of industrial time series data, this article aims to explore more effective methods for imputing missing values in time series data, in order to improve data quality and provide more reliable data support for decision analysis in the industrial field.

II. APPLICATION DISCUSSION OF THE CASE

This case study follows a rigorous scientific research approach. Firstly, we provide a comprehensive overview of the data sources and application background of industrial temporal data scenarios, clarifying the practical significance and potential value of the study. Next, we delved into the proposed solution and elaborated on its conceptual design, overall framework, and specific algorithm description. In this process, we strive to ensure the scientificity, rationality, and operability of the plan. After completing the theoretical

Manuscript received May 24, 2024

Peiyuan Yang, School of Software, Tiangong University, Tianjin, China
Jin Hao, Boya Chuangzhi (Tianjin) Technology Co., LTD, Tianjin, China

construction, we further designed experimental procedures to verify the performance and advantages of the proposed solution. Through a series of carefully designed experiments, we demonstrated the practical effectiveness of this scheme in filling missing values in industrial time series data, verifying its effectiveness and reliability. During the discussion phase, we particularly focused on the application scenarios of filling missing values in industrial time series data and analyzed the potential and advantages of this solution in solving real-world problems. This discussion further demonstrates the practical value and importance of our research.

III. Design Philosophy of the Case

The core design concept of this case study is to develop an efficient missing value filling method targeting the unique characteristics of industrial time series data. This method aims to address the problem of varying degrees of data missing, and utilizes advanced technologies such as autoencoders and generative adversarial networks to generate missing data points. In the process of model construction, we used a representative industrial time series dataset and applied completely random missing processing to simulate incomplete data in real industrial environments. Subsequently, we used the designed model to accurately and effectively fill in missing values, thereby verifying the superiority and practicality of this solution in solving the problem of missing industrial temporal data.

In this case study, we selected the meteorological dataset from the KDD Cup 2018 Challenge (hereinafter referred to as KDD) as the original dataset[7]. This dataset is sourced from meteorological observations in the Beijing area and covers sensor data collected by multiple scattered meteorological observation stations throughout Beijing. In order to explore the impact of missing data on the quality of meteorological data, we specifically selected data from 11 representative meteorological observation stations, whose datasets include 12 key meteorological attributes such as PM2.5 concentration, carbon monoxide concentration, and temperature.

In order to simulate possible data missing situations in real environments, we applied varying degrees of complete random missing processing to the original dataset, with the missing proportion gradually increasing from 10% to 90%. Subsequently, we applied the designed model to efficiently fill in missing values on these datasets with missing values. Through this carefully designed series of experiments, we aim to verify the performance of the proposed model in dealing with varying degrees of data loss, as well as its potential in improving data integrity and accuracy.

IV. Algorithm Description of the Case

A. *Data denoising module*

Given that the original dataset may have some noise and redundancy due to various external factors (such as sensor errors, environmental factors, etc.), these interference factors may not only affect the accuracy and reliability of the data, but also have adverse effects on subsequent data analysis and model training. In order to improve data quality, we use a denoising autoencoder to preprocess the raw data.

Denoising autoencoder is a deep learning architecture that is particularly suitable for processing datasets containing noise[8]. The basic principle is to introduce noise into the input data, and then train an autoencoder network to recover the original noise free data from the noisy data, as shown in Figure 1..

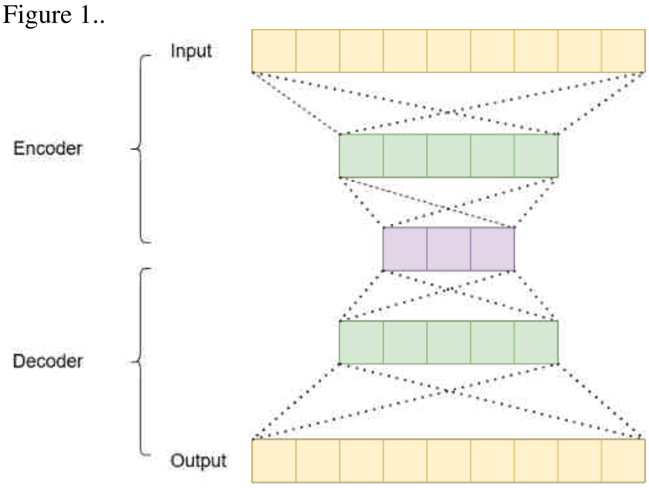

FIG. 1 Schematic diagram of denoising autoencoder structure

Denoising autoencoder consists of two core components: encoder $f_\varphi(x)$ and decoder $g_\theta(y)$. The encoder is responsible for non-linear transformation of the original input data x through the network, extracting key features from the data, and generating a low dimensional latent representation y, which is commonly referred to as encoding or hidden state. This process is shown as follows:

$$y = f_\varphi(x)$$

The decoder is another key component of the auto-encoder, responsible for decoding and reconstructing the potential representation y generated by the encoder through the network. The function of the decoder is to map y back to the output as close as possible to the original input x, to ensure that the potential representation y extracted during the conversion process contains representative information that can represent the input data. This process is shown as follows:

$$x' = g_\theta(y) = g_\theta(f_\varphi(x))$$

The feature vectors generated by the encoder are used in the denoising autoencoder to reconstruct the original input, and the basic principle of this process is to train the entire network by minimizing the reconstruction error. Specifically, the loss function is designed to measure the difference between the reconstructed output and the original input, continuously optimizing network parameters through backpropagation algorithms and gradient descent methods to minimize this loss function. The network loss function is shown as follows:

$$L = \|x - x'\|_2^2 = \|x - g_\theta(y)\|_2^2 = \left\|x - g_\theta(f_\varphi(x))\right\|_2^2$$

Among them, φ is the network parameter of the encoder, and θ is the network parameter of the decoder.

B. *Missing data generation*

After denoising the original dataset, we will now use generative adversarial networks, an advanced deep learning technique, to generate missing data. Generative adversarial networks consist of two main neural networks: a generator and a discriminator. The task of the generator is to generate new, seemingly real data based on known data distributions, while the discriminator is responsible for determining whether the input data comes from a real dataset or is generated by the generator[9].

In this case, we use a generator to generate missing meteorological data. Specifically, the generator learns the features and patterns extracted from the original dataset, and based on this information, simulates and generates missing data points. At the same time, the discriminator continuously evaluates the data generated by the generator to ensure that these data are consistent with real data in terms of statistical characteristics and patterns.

In the training process of generating adversarial networks, a dynamic game process is presented, with the goal of pursuing a Nash equilibrium state. The essence of this game is achieved by optimizing a specific objective function, which is described as follows:

$$min_G max_D V(D,G) = \mathbb{E}_{x\sim p_x}[logD(x)] + \mathbb{E}_{x\sim p_z}[log(1 - D(G(z)))]$$

Among them, $p_x(x)$ represents raw data, $p_z(x)$ represents random noise, $D(x)$ represents inputting raw data into the discriminator network, $G(x)$ represents inputting random noise into the claimed network, and $D(G(x))$ represents inputting the output of the generator into the discriminator and scoring. The training goal of discriminator network D is to make $logD(x)$ and $log(1 - D(G(x)))$ as large as possible, that is, to maximize the score of the original data and minimize the score on the generated data. The training goal of the generator network is to make $log(1 - D(G(z)))$ as small as possible, that is, to maximize the loss of the discriminator network. Ultimately, the output of the generator will be closer to the real sample.

From this, it can be concluded that in the generative adversarial network, the loss functions of the discriminator and generator are as follows:

$$G_{loss} = log[1 - D(G(z))]$$

$$D_{loss} = log[1 - D(G(z))] - logD(x)$$

If the loss function value of generator network G is smaller, i.e. $D(G(z))$ is closer to 1, it indicates that the discriminator gives a higher rating to the data output by the generator; If the loss function value of discriminator network D is smaller, $D(x)$ is larger, and $D(G(z))$ is smaller, it indicates that the discriminator is better able to distinguish between the output data of the generator and the original data.The process is shown in Figure 2:

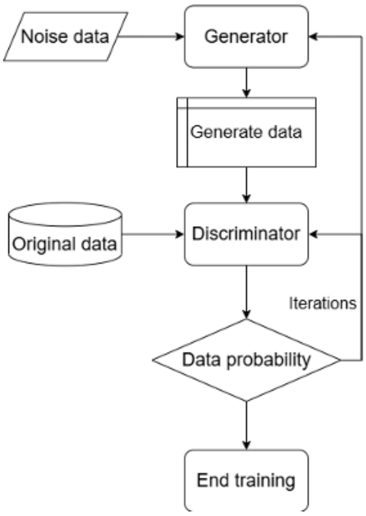

FIG. 2 Flowchart of Generative Adversarial Network

V. Experimental Analysis of the Case

A. Experimental Setup and Dataset

In the experimental design of this case study, in order to ensure the accuracy and stability of the model performance, we adopted the classic evaluation method of five fold cross validation. The strategy of dividing the original dataset into five equally large subsets is to select four of them as the training set and the remaining one as the validation set each time. This process will be repeated five times, each time selecting a different subset as the validation set to ensure that each subset has a chance to be used for validation.

Through this method, we can obtain five different model performance evaluation results. In order to obtain a more representative performance indicator, we will average these five results, calculate the average value and possible error range. This method not only improves the reliability of model evaluation, but also helps us identify potential overfitting or underfitting issues, thereby optimizing the model construction and parameter tuning process.

B. Evaluation Methods and Metrics

In order to directly and accurately quantify the performance of the proposed model in missing value filling tasks, this case study used Root Mean Square Error (RMSE) as the evaluation metric. RMSE is a commonly used performance measurement method for regression problems, which can measure the degree of difference between predicted values and true values, and decouple the magnitude of error from data size by taking the square root, making the error measurement of different datasets more comparable. In this case, we will compare the filled data with the missing parts of the original data and calculate its RMSE value.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}[\hat{y_i} - y_i]^2}$$

C. Experimental Results and Analysis

In order to comprehensively evaluate the performance of the missing value filling method proposed in this study, we selected two classic missing value filling methods as comparison benchmarks, namely K-nearest neighbor algorithm (KNN) and traditional generative adversarial networks (GANs). The comparison results are shown in Figure 3.

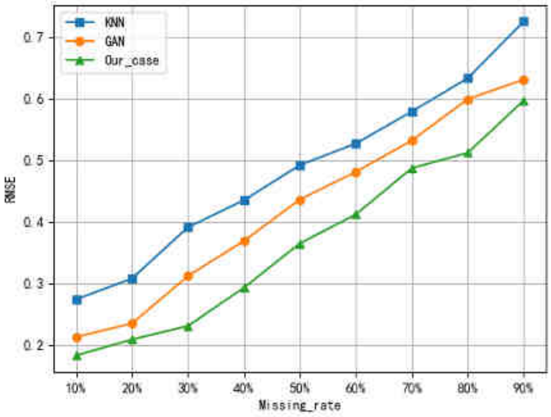

FIG. 3 Comparison of RMSE under different missing rates using different methods

After in-depth analysis of the experimental results, we can draw the following conclusions. Firstly, The KNN method is inadequate when dealing with high-dimensional and complex temporal datasets. This is because the essence of KNN algorithm is to find similar samples based on the distance between samples, and then predict missing values. However, in high-dimensional space, the distance measurement between samples becomes complex and unstable, making it difficult for KNN to accurately capture the intrinsic correlations and patterns between data. In addition, the presence of noise and redundant information in the original dataset further exacerbates the performance degradation of the KNN algorithm in missing value filling tasks. On the other hand, traditional GANs also face challenges when directly applied to industrial time series data. Although GANs perform well in generating new data, they lack necessary preprocessing steps when processing data in specific fields, especially for data with complex structures and patterns such as industrial time series data. This makes it difficult for GANs to fully learn the inherent patterns and features of the data, thereby affecting their performance in missing value filling tasks.

In contrast, the method proposed in this case study demonstrated good performance in experiments. This is mainly due to our preprocessing steps such as denoising and dimensionality reduction on the original data, effectively improving the quality and availability of the data. At the same time, the model designed in our case can fully utilize the temporal characteristics and patterns of industrial temporal data, and generate high-quality missing data through generative adversarial networks, thereby achieving accurate filling of missing values. This result not only verifies the effectiveness and superiority of the method proposed in this study, but also provides new ideas and solutions for filling missing values in industrial time series data.

VI. Conclusion

In order to solve a series of problems caused by missing sensor data in industrial environments, this paper proposes an innovative method based on denoising autoencoder and generative adversarial network, aiming to achieve accurate filling of missing values in industrial sensor data. We introduced the basic theories, technical approaches, and evaluation methods of denoising autoencoder (DAE) and generative adversarial network (GAN). Taking the KDD Cup 2018 dataset as an example, a simulated industrial environment was constructed using this dataset to evaluate the accuracy of missing data filling and filling. The experimental results show that our proposed method performs well in missing value filling tasks, not only accurately filling missing values, but also has high data quality after filling. It can be widely applied in sensor data analysis tasks in various industrial scenarios.

References

[1] Da Xu, Li. "Industrial information integration–An emerging subject in industrialization and informatization process." Journal of Industrial Information Integration 17 (2020): 100128.

[2] Santos, Miriam Seoane, et al. "Generating synthetic missing data: A review by missing mechanism." IEEE Access 7 (2019): 11651-11667.

[3] Goodfellow, Ian, et al. "Generative adversarial networks." Communications of the ACM 63.11 (2020): 139-144.

[4] Yoon, James Jordon, and Mihaela Schaar. "Gain: Missing data imputation using generative adversarial nets." International conference on machine learning. PMLR, 2018.

[5] Shang, Chao, et al. "VIGAN: Missing view imputation with generative adversarial networks." 2017 IEEE International conference on big data (Big Data). IEEE, 2017.

[6] Luo, Yonghong, et al. "Multivariate time series imputation with generative adversarial networks." Advances in neural information processing systems 31 (2018).

[7] CUP KDD. Available on: https://www.kdd.org/kdd2018/

[8] Dong, Yafen, **aohong Shen, and Haiyan Wang. "Bidirectional denoising autoencoders-based robust representation learning for underwater acoustic target signal denoising." IEEE Transactions on Instrumentation and Measurement 71 (2022): 1-8.

[9] Deng, Grace, Cuize Han, and David S. Matteson. "Extended missing data imputation via GANs for ranking applications." Data Mining and Knowledge Discovery 36.4 (2022): 1498-1520.