Transformer-GRU-GAN Based Data Restoration Approach

Jun Tang[1,3], Bing Guo[1], Yan Shen[2], Shengxin Dai[1], Yuming Jiang[1]*

[1] College of Computer Science, Sichuan University, China
[2] School of Computer Science, Chengdu University of Information Technology, China
[3] Changhong Central Research Institute, Sichuan Changhong Electronic (Group) Co., Ltd., China
9535390@qq.com, guobing@scu.edu.cn, sheny@cuit.edu.cn, daishengxin@scu.edu.cn, jiangym@scu.edu.cn

Abstract

The integrity of data greatly affects its quality, and thus its asset value and also the accuracy of data analysis tasks based on it. However, in reality, there are often various factors that cause data deficiency. Although many scholars have made researches on this problem, most of the studies have limited accuracy. Therefore, this paper proposes a transformer-GRU based GAN, which first learns the potential distribution of real data through generator formed by stacked encoder of transformer to generate the complementary data, and then inputs the complementary data into the discriminator based on GRU for discrimination. Finally, it trains the model through the adversarial game between generator and discriminator. Comparison experiments are carried out based on the sales data of a company, and it can be seen that the evaluation indicators of the method proposed in this paper are superior to those of the compared methods, proving that the missing data recovery method proposed in this paper has better accuracy.

Keywords: Data completion, Generative adversarial network, Transformer, Gated recurrent unit

1 Introduction

Since January 1st of 2024, the 'Provisional Regulations on Accounting Treatment of Enterprise Data Resources' issued by the Ministry of Finance, have been in effect, advancing a key step of data towards capitalization as the fifth major factor of production, indicating that data has an important asset value. Furthermore, since important information can be mined from data to provide important references to enterprises, governments, etc. for decision-making, it is indicated that data has important application value as well. However, in practice, data may face interference from a variety of factors at various stages, i.e., collection, transmission, storage, etc. Situation such as the transmission link instability can result in missing and garbled data, significantly impacting the data quality, and consequently, affecting the prediction accuracy and other subsequent data mining tasks. Thus, the study of the missing data problem is of great significance, which can not only improve the accuracy of the subsequent research tasks, but also enhance asset value of data.

Till now, several scholars have researched the problem of missing data, but most of the models proposed have low accuracy, making them difficult to be applied in practice. Therefore, in this paper, by leveraging GAN's ability in learning potential distribution of real data, transformer's multi-attention mechanism, and GRU's ability in efficient information learning of time-series, we construct a transformer-GRU based GAN. Furthermore, by introducing time decay factor and scaling factor in attention mechanism and residual connection process, respectively, we improve the efficiency and accuracy of the model.

This method first calculates the corresponding mask matrix and the random noise matrix of missing bits based on missing data. Secondly, the missing data is 0-filled and input into the generator formed by stacked encoder of transformer along with the mask matrix and noise matrix. Then, the complementary data generated is input into the GRU-based discriminator. Finally, an adversarial game is held between generator and discriminator. Loss is back propagated and model parameters are updated. Comparative experiments based on a company's sales data are conducted to verify the effectiveness of method proposed in this paper.

The main contributions of this paper are as follows:

1) The encoder of Transformer is improved. Aiming at the characteristics of time-series data, a time decay factor is introduced to improve the multi-head self-attention mechanism by realizing the adjustment of attention weights. Meanwhile, a scaling factor is added in the process of residual connection and layer normalization, which effectively balances the inputs of corresponding layer, thus improving the stability and generalization ability of the model.

2) The generator is designed based on the above optimized encoder, and combined with the GRU-based discriminator, a GAN network is constructed, which is applied to the data recovery task to improve data recovery accuracy.

3) A comparative experiment is conducted based on a company's sales data. The experimental results indicate that compared with existing methods, the method proposed in this paper not only possesses higher data recovery accuracy, but also shows strong robustness. Specifically,

*Corresponding Author: Bing Guo; Email: guobing@scu.edu.cn
DOI: https://doi.org/10.70003/160792642026072704002

as the rate of missing data increases, the error increase of the proposed method remains relatively small, providing strong support for the practical application of the proposed method in data recovery.

The remainder of this paper is organized as follows: Section 2 presents the relevant literature. Section 3 describes the various steps of the data completion algorithm proposed in this paper as well as the overall algorithmic flow, and briefly describes the underlying algorithms the proposed model relies on for a better understanding of the data completion algorithm proposed in this paper. Section 4 introduces the related data experiments and analyzes the results, taking the real sales dataset as the experimental object to verify the effectiveness of the data complementation algorithm proposed in this paper, which is mainly involved in the construction of the experiment environment, the preparation and preprocessing of the data, the selection of model hyperparameters and model performance evaluation indexes, and the training of the model. Finally, in Section 5, the main research results and contributions achieved in this paper are outlined, and the future research direction is initially discussed.

2 Literature Review

Currently, there are several ways to deal with the problem of missing data [1-2], but since simple deletion and direct neglect will affect the amount of data information, data repair has become a mainstream processing method. A number of researchers have proposed methods for data restoration, leveraging both spatial and temporal correlation in data. These methods can be mainly divided into two categories.

The first category is the data restoration models constructed based on traditional statistical or mathematical methods. For example, in the orientation of restoring data distribution, Yu et al. proposed an interpolation method that minimized maximum mean difference and variance between the interpolated values and the original complete values [3]. Zhang et al. improved the sparse basis of compressed perception to achieve effective data restoration [4]. Yang et al. firstly split the high-dimensional data into multiple low-dimensional data according to the correlation between variables, and then completed the data to reduce the impact of irrelevant and low-correlation data on the filling accuracy [5]. Li et al as well as Ma et al used probabilistic principal component analysis to repair missing data for structural health inspection [6-7]. Ji et al. proposed a Pearson correlation coefficient interpolation method based on similar moments to repair data of the power station [8].

The second category is data repair models built based on machine learning. For instance, Zhang et al proposed a data restoration method based on Bayesian dynamic regression and demonstrated that the multivariate quadratic Bayesian dynamic regression model has higher data reconstruction accuracy in experiments using architectural model data and bridge data [9]. Li et al. and Miao et al. both proposed data restoration methods based on support vector machines (SVM) [10-11]. Li et al. employed SVM to fill the multivariate time series, while Miao et al. integrated support vector regression (SVR) with the simple mean method and dynamical selection of explanatory variables to propose their data restoration method. This comprehensive approach considers various characteristics of traffic detector data, including temporal and spatial correlation, cyclical tendency, and real-time variability. Su et al., Xu et al., Cao et al. and Yu et al. proposed missing value interpolation methods based on the nearest neighbor algorithm [12-16], in which Su et al. improved the nearest neighbor algorithm by using the localization-preserving projection and sparse self-coding, respectively, and Xu et al. employed the nearest neighbor algorithm based on the Mahalanobis distance. Sun et al. combined neural networks and Gaussian processes to complete missing data of time series in the context of small datasets and applied it to the Beijing PM2.5 content dataset as well as the ocean surface temperature dataset to test the algorithm performance [17]. Bu et al., Chen et al. and Xu et al. all proposed data repair models based on self-coding models [18-20]. Among them, Bu et al. first clustered datasets using the fast density clustering algorithm, and then filled the missing data in chunks using the improved noise-reducing self-coding model. Chen et al. extracted the deep correlation relationship between data using the deep noise-reducing self-coding network, and predicted the data to be repaired with the SVR, and verified the performance based on the noise data of an airport. Xu et al. extracted spatio-temporal features of the missing data in road network through a noise-reducing graph variational self-encoder, and based on this extracted spatio-temporal features, they used a generative adversarial network to generate the road network data to achieve interpolation of missing data, and carried out the comparative experiments of data restoration under different types of missing and missing rates. Hu et al proposed a hierarchical data recovery method based on generative adversarial networks for missing pressure data in pipeline network [21]. Zhang et al., Wang et al., Hou et al., Chai et al., Tang et al. and Lin et al. proposed data repair methods based on convolutional network [22-27]. Among them, Wang et al. and Hou et al. applied the proposed methods to the restoration of missing wind speed data and missing traffic flow data, respectively. Lin et al. combined the signal decomposition and interpolation methods and applied it to the recovery of missing wind speed data. Lu et al. established a fault data repair model based on gray-wavelet neural network [28]; Dai et al. utilized symmetric residual U-network for repairing traffic flow data of road network [29]. Shao et al. and Li et al. proposed traffic flow data restoration methods based on random forests [30-31]. Between them, the former also used genetic algorithm to optimally tune the parameters of random forest model. Yu et al. and Wang et al. proposed missing data repair methods based on Long Short-term Memory (LSTM) networks [32-33], and Yu et al. also incorporated a multiple attention mechanism. Wang et al. proposed a data filling method based on generalized central clustering to fully employ the useful information of partially missing data [34]. Liu et al. proposed a

transformer method based on convolutional network and self-attention mechanism to repair the data [35]. Yang et al. first mapped one-dimensional power quality data into two-dimensional grayscale images, and then applied fuzzy self-organizing neural networks to cluster the data. Finally, they restored the clustered data hierarchically [36]. Dai et al. proposed a tensor ring completion method based on Bayesian inference, which can automatically learn the optimal rank for tensor ring completion to effectively recover missing data in intelligent transportation systems [37]. Dai et al. and Lu et al. proposed improved tensor completion methods based on nuclear norm and truncated nuclear norm, which leverages prior rank information and retains large/key singular values to optimize the recovery of missing traffic data and video data respectively [38-39].

3 The Proposed Approach

Generative adversarial network (GAN) is a neural network designed on the basis of game theory, which consists of two parts: generator and discriminator. In the training process, the generator generates false data, and the discriminator judges between the real and false data. The generator continuously generates false data that is close to real data to deceive the discriminator, and the discriminator continuously improves its own discriminatory capability. According to this adversarial training, model parameters are continuously adjusted to improve the model performance and accuracy. Finally, the purpose of passing off the spurious as genuine is achieved, which allows the model to generate the required data. Structure of GAN is shown in Figure 1. To solve the problem of missing data recovery, this paper designs a data restoration model based on GAN.

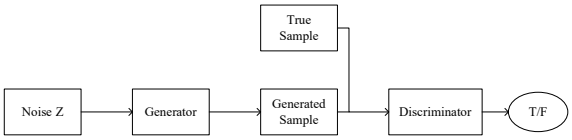

Figure 1. Structure of GAN

3.1 Generator

Transformer is an advanced model for processing sequential data in the field of Natural Language Processing (NLP). It was initially introduced for machine translation tasks, but had since been widely used for other generative tasks as well. It introduces an attention mechanism to dynamically adjust the weights of input data, and employs residual concatenation and layer normalization for better performance. Transformer model is obtained by stacking multiple Encoders and Decoders, where Encoders are used to learn representations of input sequences, and Decoders use outputs of Encoders and context information to generate output sequence. Generators for traditional GAN are usually designed based on Convolutional Neural Networks (CNN) or Recurrent Neural Networks (RNN). The generator designed in this paper, however, uses encoder of Transformer. The structure of a single encoder of Transformer is shown in Figure 2.

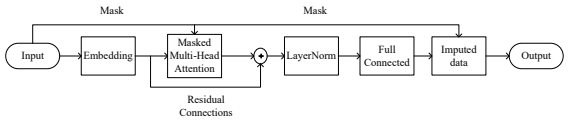

Figure 2. Structure of encoder

For an input sequence with time step n and number of features m, its matrix X is represented as shown in equation (1), where X is an $n\times m$ matrix and x_{ij} is the j-th feature at the i-th moment.

$$X^{n\times m}=\begin{pmatrix} x_{11}\,x_{12}\ldots x_{1j} \\ x_{21}\,x_{22}\ldots x_{2j} \\ \vdots\ \vdots\ \vdots\ \vdots \\ x_{i1}\,x_{i2}\ldots x_{ij} \end{pmatrix} \tag{1}$$

Transformer's encoder adds positional features to the input sequence through positional embedding in the embedding layer, which helps the model to understand relative positions of different time steps and learn positional features of sequences. The position embedding is a positional embedding matrix obtained by periodic sine-cosine function. Such position embedding matrix is calculated as shown in equations (2) and (3), where i and j denote the j-th position of the i-th sequence and m denotes the dimensionality. The position embedding matrix has the same dimension as the input sequence X, so that it can be added to the input sequence element by element to obtain the position embedded matrix X_1, the process is shown in equation (4).

$$P_{i,2j}=\sin\left(\frac{i}{10000^{\frac{2j}{m}}}\right) \tag{2}$$

$$P_{i,2j+1}=\cos\left(\frac{i}{10000^{\frac{2j}{m}}}\right) \tag{3}$$

$$X_1=X+P \tag{4}$$

After the action of positional embedding matrix, the input sequence is mapped through a series of fully connected layers to transform features at each time step nonlinearly. Thus, complex relationships and feature representations in sequence are captured. The corresponding calculation is shown in Equation (5), where W_1, b_1 are parameters of the first fully connected layer, W_2, b_2 are parameters of the second fully connected layer, σ is the *ReLu* activation function, and $X_2(X_2\in R^{n\times m})$ is the embedding layer's output.

$$X_2 = \sigma\left(\left(W_1 X_1 + b_1\right) W_2 + b_2\right) \tag{5}$$

To enhance the representation ability of model, the encoder introduces a multi-head attention mechanism. After each attention head learns a different relevance weight, they are merged to obtain the final attention score. The relationship between different time steps in the sequence may be different, and weaker dependencies may exist between more distant time steps. Therefore, a time decay self-attention mechanism is introduced. According to the adjustment of attention weights by time decay factor, temporal preferences and correlations are taken into account. The time decay factor matrix is calculated as shown in equation (6), where $\eta_{i,j}$ is the decay coefficient at the j-th position of the i-th sequence, w is the learnable periodic coding frequency parameter, and *sin* is the sine function.

$$\eta_{i,j} = \sin\left(w * \left(\left|i - j\right|\right)\right) \tag{6}$$

For the i-th moment, if the j-th feature is missing, it is labelled as 0, otherwise labelled as 1. On the basis of this rule, a mask matrix M is constructed and M is homomorphic with X. An example is shown in equation (8), where 'Null' is the missing value in input sequence.

$$M_{ij} = \begin{cases} 0 \text{ if } X_{ij} \text{ is absent} \\ 1 \text{ if } X_{ij} \text{ is present} \end{cases} \tag{7}$$

$$\begin{cases} X^{2\times4} = \begin{bmatrix} 1 & 2 & Null & Null \\ 2 & Null & 3 & 0 \end{bmatrix} \\ M^{2\times4} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \end{cases} \tag{8}$$

The calculation of the self-attention mechanism defined by combining mask matrix and time decay factor matrix under single-head attention mechanism is shown in equation (9). In this equation, Q, K and V are query matrix, key matrix and value matrix separately, which are obtained by linear transformation of input sequence. $1/\sqrt{d_k}$ is scaling factor and *softmax* is normalization process.

$$Attention\left(Q, K, V\right) = softmax\left(\frac{QK^T}{\sqrt{d_k}} \eta M\right) V \tag{9}$$

The number of multi-head attention mechanism heads is h. A richer and more integrated representation matrix X_3 is obtained after the concatenation and linear transformation of the results affected by each attention mechanism head, and is used as input of the next layer. The process is shown in equation (10), and the calculation of the i-th attention head is shown in equation (11), where W^o, W_i^Q, W_i^K, W_i^V are learnable weight parameter matrices and *Concat* is concatenation operation.

$$X_3 = Concat\left(head_1, \ldots, head_h\right) W^o \tag{10}$$

$$head_i = Attention\left(X_2 W_i^Q, X_2 W_i^K, X_2 W_i^V\right) \tag{11}$$

The multi-head self-attention mechanism is followed by residual connection and layer normalization. The residual connection adds the current input sequence directly to the embedding layer output and then performs normalization operation through layer normalization, which helps to mitigate gradient vanishing as well as increases training speed. In order to better balance the inputs of residual connection, a scaling factor ε is introduced and the process is shown in equation (12).

$$X_4 = LayerNorm\left(\varepsilon\left(X_3 + X_2\right)\right) \tag{12}$$

Finally, the results of a single encoder are output through a fully connected layer as shown in equation (13), where W_3, b_3 are learnable parameters of the fully connected layer and σ is the *ReLu* activation function.

$$X_5 = \sigma\left(W_3 X_4 + b_3\right) \tag{13}$$

The generator is designed based on above encoder structure, and utilizes a stack of two encoders. For the randomness and diversity of the data generated by generator, random noise z satisfying normal distribution is introduced between encoder layers, where z and X_5 are homogeneous matrices. z is applied to the output X_5 of the first layer and then inputs to the second encoder layer to get the result after filling. Based on mask matrix, the complementary data for the missing positions is calculated and filled to the true sequence X, resulting in the complementary data X'. This process is shown in equation (14).

$$\begin{aligned} X' = Encoder\left(Encoder\left(X\right) + z\right) \\ \odot\left(1 - M\right) + X \odot M \end{aligned} \tag{14}$$

The complementary data generated by the generator will be fed into discriminator, which will be used to discriminate the authenticity of input data. This interaction process between generator and discriminator forms the core of the game relationship in GAN.

3.2 Discriminator

Gated Recurrent Unit (GRU), as a variant of RNN, benefits from the memory of past information and is

commonly used in processing sequence tasks. Compared to LSTM, GRU occupies simpler structure, faster convergence speed, and can also solve the long-term dependency problem of RNN due to the unique design of update gate and reset gate structure. The update gate determines how much the current information is affected by the previous information, and the reset gate determines whether the information of previous step is ignored. The training process of GAN is usually unstable and prone to pattern collapse or pattern oscillation. As a discriminator, GRU can provide a certain degree of stability. It can capture temporal information of input samples, and thus offer a more comprehensive discriminative ability. The discriminator is implemented using GRU with an output dimension of 1, which outputs the probability that input sample is the real data. The discriminator will give a higher probability for the real data and a lower probability for the complementary data generated by generator, thus achieve the discrimination of authenticity of data. The structure of GRU is shown in Figure 3.

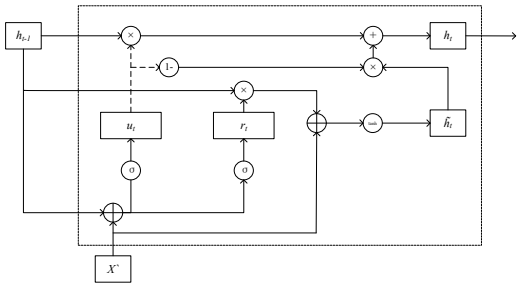

Figure 3. Structure of GRU

The update process of GRU is shown in equations (15) to (18), where u_t is the value of update gate, h_{t-1} is the hidden state of previous step, r_t is the value of reset gate, σ is *sigmoid* activation function, $\tilde{h}_t$ is the candidate of current hidden state, h_t is the current hidden state, and W_u, W_r, $W_{\tilde{h}}$, b_u, b_r and $b_{\tilde{h}}$ are trainable parameters. This structure allows GRU to efficiently control the current inputs and outputs through the gate structure for more effective processing of sequences.

$$u_t = \sigma\left(W_u \times \left[X', h_{t-1}\right] + b_u\right) \tag{15}$$

$$r_t = \sigma\left(W_r \times \left[X', h_{t-1}\right] + b_r\right) \tag{16}$$

$$\tilde{h}_t = \tanh\left(W_{\tilde{h}} \times \left[r_t \odot h_{t-1}, X'\right] + b_{\tilde{h}}\right) \tag{17}$$

$$h_t = \left(1 - u_t\right) \odot h_{t-1} + u_t \odot \tilde{h}_t \tag{18}$$

3.3 Algorithm Flow

The generator of GAN learns the distribution of true sequence to generate fake data that is similar to true sequence, and the discriminator discriminates this generated data. The two networks fight against each other and adjust parameters during the training process, which eventually makes discriminator unable to distinguish between the data generated by generator and the real one. Loss of generator G_{Loss} is shown in equation (19) and that of discriminator D_{Loss} is shown in equation (20), where $G(\cdot)$ is the generator, $D(\cdot)$ is the discriminator, and λ is coefficient of reconstruction loss. For discriminator, it is required that the probability $D(X)$ of discriminating on true sequence to be as close to 1 as possible, while the probability $D(G(X))$ of discriminating on generated data to be as close to 0 as possible. For generator, it is required that the probability $D(G(X))$ of discriminating on generated data to be as close to 1 as possible.

$$G_{Loss} = -D\left(G\left(X\right)\right) + \lambda \left\| X - G\left(X\right) \right\|_2 \tag{19}$$

$$D_{Loss} = -D\left(X\right) + D\left(G\left(X\right)\right) \tag{20}$$

In training process, firstly, the missing part of true sequence X is filled with 0, and the corresponding mask matrix M and noise z are constructed. The true sequence after 0-filling is min-max normalized, and input to generator to generate the corresponding complementary sequence X'. According to true sequence and complementary sequence, loss of generator G_{Loss} is calculated. The complementary data is input to discriminator, and loss of discriminator D_{Loss} is calculated. Then, feedback of D_{Loss} is given to discriminator and the sum of G_{Loss} and D_{Loss} to generator, so that the generator and discriminator fight against each other to get the final sequence which approximates the true data distribution to achieve missing data recovery. The whole algorithmic framework is shown in Figure 4.

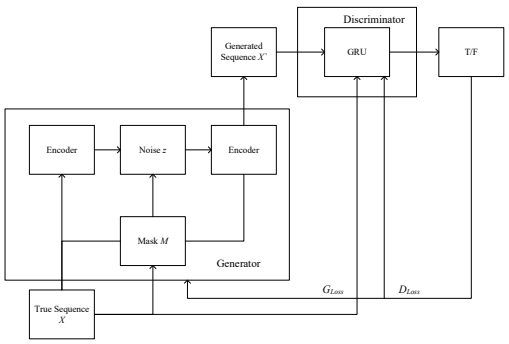

Figure 4. Algorithmic framework

Algorithm 1. Data restoration algorithmic

Input: Missing sequence X

Output: Complementary sequence X'

Algorithm:

Initialize hyperparameters of model

while epoch does not exceed the maximum value **do**:

Based on input sequence X, compute mask matrix M.

Based on M, compute noise matrix of missing bits z.
After 0-filling and min-max normalization of X, it is fed into generator together with M and z.
Generator generates complementary sequences X'.
Calculate the loss of generator G_{Loss}.
Input X' into discriminator.
Calculate the loss of discriminator D_{Loss}.
Backpropagate and update parameters.
end while

4 Experiments

4.1 Experimental Data and Experimental Environment

In this paper, real sales data from a company is used to validate model performance. The dataset consists of sales data from 133 shops in 172 weeks from 2016 to 2019. The data contains in total 27 features such as record date, sales amount, number of items sold, customer flow, number of VIPs, conversion rate of customers, total number of items sold for each shop, etc. 2% of the data is missed, which is a low missing rate and it can be basically determined that the data belongs to a complete dataset. Some data samples are shown in Table 1. Designing experiments using a complete dataset allows for better validation of the model's effectiveness in repairing missing data. Missing data scenarios are simulated by randomly replacing feature values with 'Null'.

Construction of network together with training and testing processes of the experiments in this paper are implemented based on pytorch framework. Adam optimizer serves as a relative mainstream choice in sequence prediction due to its good performance. For this reason, it is used to optimize the training process in this paper. The experiment environment is shown in Table 2.

Table 1. Data samples

Date	Shop ID	Number of VIPs (person)	Quantity sold of commodity A (piece)	Quantity sold of commodity B (piece)	Total quantity sold (piece)	…	Sales (CNY)
2016/01/01	01	19	157	133	425	…	164195.74
2016/01/08	03	45	223	124	658	…	197251.28
2016/01/15	02	23	125	79	387	…	159496.81
2016/01/22	05	21	164	102	326	…	149425.15

Table 2. Experiment environment

System environment	Windows 10 64BIT
GPU	NVIDIA GeForce RTX3070
Programming language	Python 3.8
Development tool	Pycharm
Anaconda	Anaconda 4.12.0
CUDA	Cuda 11.1
Pytorch	1.9.0

4.2 Evaluation Indicators and Hyperparameters Value

In order to verify the model's accuracy of data completion, Mean Square Error (MSE) and Mean Absolute Error (MAE) are used as the evaluation indicators, as shown in equations (21) and (22), where y is the true value, y' is the predicted value and n is the number of samples.

$$MSE = \frac{\sum(y - y')^2}{n} \tag{21}$$

$$MAE = \frac{\sum|y - y'|}{n} \tag{22}$$

Following a series of experiments, the primary hyperparameters for model training are set as in Table 3.

Table 3. Hyperparameter value

Hyperparameter	Meaning	Value
epoch	Number of iterations	1000
lr	Learning rate	0.001
hidden_size	Size of hidden units	128
weight_decay	Regularization parameter	0.001
h	Attention heads	8

4.3 Comparative Experiment

In order to verify the performance of the model for missing data repair, the original dataset is treated as a missing dataset by replacing operation of 'Null' in the proportion of 5%, 10% and 15%, respectively, and then the corresponding mask matrix is obtained. After the simulation of missing dataset, 'Null' is replaced with 0 and min-max normalized preprocessing is performed. All the data is divided 8:2 on a total basis of shops' number, with 107 shops as training data and 26 shops as testing data.

Commonly used missing value processing methods usually include direct deletion, mean interpolation (MI), median interpolation, k-nearest neighbor algorithm (KNN), least squares, matrix factorization (MF), etc. In this paper, MF, MI, KNN, and GRU-GAN are taken as comparison methods for experiments. MF decomposes the original data matrix with missing values into the product of two low-rank matrices to fill in missing values and to get complete data. KNN filling uses clustering to obtain a

number of sample points in the neighborhood of a given missing sample, and performs missing value filling by calculating the mean or weighted average of these sample points. GRU-GAN is a GAN where both generator and discriminator are implemented using the GRU network. Results of comparative experiments are shown in Table 4 to Table 6.

Table 4. Results of comparative Experiments at 5% of missing values

Indicator \ Method	MF	MI	KNN	GRU-GAN	Ours
MSE	0.0736	0.0661	0.0521	0.0329	0.0261
MAE	0.0577	0.0540	0.0457	0.0308	0.0194

Table 5. Results of comparative Experiments at 10% of missing values

Indicator \ Method	MF	MI	KNN	GRU-GAN	Ours
MSE	0.0947	0.0722	0.0606	0.0427	0.0332
MAE	0.0772	0.0661	0.0551	0.0399	0.0262

Table 6. Results of comparative Experiments at 15% of missing values

Indicator \ Method	MF	MI	KNN	GRU-GAN	Ours
MSE	0.103	0.0821	0.0693	0.0521	0.0413
MAE	0.094	0.0703	0.0661	0.0547	0.0358

From the experimental results, it can be concluded that the performance of each algorithm decreases as the missing rate increases. Among them, MF and MI perform worse in learning of overall data patterns with increase of missing data. As a result, if data is poor in completeness, accuracy of data repair will be low with the usage of these two methods. KNN has a better performance because of its neighbors' similarity. It behaves well in the case of low missing rate, and performs stable because it has relative low decrease in capability as the missing rate increases. GRU, which is realized on the basis of RNN, has good memory ability for time series data. This allows it to better process sequence data and learn its regularities. The experimental results of proposed method in this paper are superior to those of compared methods in all missing rate cases. The attention mechanism of transformer can capture correlation information between different positions in sequence, and GAN uses potential distribution information in training data to generate reasonable estimates of missing value, making filling results more realistic and accurate.

5 Conclusion and Future Research

Aiming at the problem of missing data, this paper designs a data repair model using Transformer and GRU based on the thought of GAN. Firstly, based on the structure of Transformer's encoder, for time series, the time decay factor is introduced to improve the multi-head self-attention mechanism to achieve the adjustment of attention weight. The scaling factor is introduced in the process of residual connection and layer normalization to balance input size. And the generator is formed by stacking improved encoder. Secondly, considering the simple structure and efficient training of GRU, the discriminator network is designed based on GRU. The proposed method is verified by conducting corresponding comparison experiments, and the results show that the designed network has optimal feasibility and accuracy for repair of the missing data in some extent.

However, there are still some deficiencies and limitations during the research of this paper. Although the repair results are relatively close to the actual values, they still need to be explored and verified in multiple data scenarios. For the discriminator, more efficient models can be further explored. As for the problems such as pattern collapse and pattern oscillation that tend to happen in the training process, improvements in network depth and activation function can be made to design GAN with better performance.

Acknowledgement

This work was supported in part by the National Key R&D Program of China under Grant No. 2020YFB1707900 and 2020YFB1711800; the National Natural Science Foundation of China under Grant No. 62262074, U2268204 and 62172061; the Science and Technology Project of Sichuan Province under Grant No. 2022YFG0159, 2023ZHCG0014, 2023ZHCG0011, 2022YFG0155, 2022YFG0157; National Natural Science Foundation of China under Grant No. 62302323; Sichuan Science and Technology Program under Grant No. 2023NSFSC1413 and 2023YFG0117; the Fundamental Research Funds for the Central Universities under Grant No. 2022SCU12077.

References

[1] H. C. Meng, S. Y. Chen, A Comparative Analysis of Data Imputation Methods for Missing Traffic Flow Data, *Journal of Transport Information and Safety*, Vol. 36, No. 2, pp. 61-67, March-April, 2018.

[2] Z. M. Xiong, H. Y. Guo, Y. X. Wu, Review of Missing Data Processing Methods, *Computer Engineering and Applications*, Vol. 57, No. 14, pp. 27-38, July, 2021.

[3] J. Y. Yu, Y. L. He, L. Z. Cui, Z. X. Huang, Distributionally Consistent Missing Value Interpolation Algorithms for Large-scale Data, *Journal of Tsinghua University (Science and Technology)*, Vol. 63, No. 5, pp. 740-753, May, 2023.

[4] J. Z. Zhang, D. Tao, Crowdsensing Based Traffic Velocity Missing Data Recovery Algorithm, *Journal of Chinese Computer Systems*, Vol. 42, No. 2, pp. 225-230, February, 2021.

[5] J. Yang, H. Yang, L. B. Wang, X. Jin, H. Guo, L. L. Yu, Research on Block Imputation Algorithm for High

Dimensional Correlation Missing Data, *Journal of Frontiers of Computer Science and Technology*, Vol. 11, No. 10, pp. 1557-1569, October, 2017.

[6] L. C. Li, H. L. Liu, H. J. Zhou, C. D. Zhang, Missing Data Estimation Method for Time Series Data in Structure Health Monitoring Systems by Probability Principal Component Analysis, *Advances in Engineering Software*, Vol. 149, Article No. 102901, November, 2020.

[7] Z. Ma, Y. Z. Luo, H. P. Wan, C. B. Yun, Y. B. Shen, F. Yu, Repair Method of Structural Health Monitoring Data Based on Probabilistic Principal Component Analysis, *Journal of Vibration and Shock*, Vol. 40, No. 21, pp. 135-141+167, November, 2021.

[8] D. Y. Ji, F. Jin, L. Dong, S. Zhang, K. Y. Yu, Data Repairing of Photovoltaic Power Plant Based on Pearson Correlation Coefficient, *Zhongguo Dianji Gongcheng Xuebao/Proceedings of the Chinese Society of Electrical Engineering*, Vol. 42, No. 4, pp. 1514-1522, February, 2022.

[9] Y. M. Zhang, H. Wang, Y. Bai, J. X. Mao, Y. C. Xu, Bayesian Dynamic Regression for Reconstructing Missing Data in Structural Health Monitoring, *Structural Health Monitoring*, Vol. 21, No. 5, pp. 2097-2115, September, 2022.

[10] Z. X. Li, F. M. Zhang, Y. Wang, Q. Tao, C. Li, Method of Missing Data Imputation for Multivariate Time Series, *Systems Engineering and Electronics*, Vol. 40, No. 1, pp. 225-230, January, 2018.

[11] X. Miao, Z. Y. Wang, Y. J. Zhou, B. Wu, Improved Modification Method of Missing Data for Location-specific Detector, *Journal of Tongji University (Natural Science)*, Vol. 47, No. 10, pp. 1477-1484, October, 2019.

[12] Y. J. Su, K. Sun, Z. Y. Deng, K. J. Yin, KNN padding algorithm based on LPP and l2,1, *Journal of Guangxi Normal University (Natural Science Edition)*, Vol. 33, No. 4, pp. 55-62, July-August, 2015.

[13] Y. J. Su, D. B. Cheng, M. Zong, L. Li, Y. H. Zhu, K-nearest Neighbor Imputation Based on Sparse Coding, *Application research of Computers*, Vol. 32, No. 7, pp. 1942-1945, July, 2015.

[14] J. Xu, Z. K. Chen, Q. C. Zhang, On a Missing Data Imputation Algorithm Based on the Nested Sliding Window, *Journal of Southwest China Normal University (Natural Science Edition)*, Vol. 40, No. 11, pp. 130-136, November, 2015.

[15] W. Q. Cao, Y. J. Chu, X. Li, Approximate Imputation Method for Missing Data in Machine Learning, *Journal of Xi'an Jiaotong University*, Vol. 51, No. 10, pp. 142-148, October, 2017.

[16] L. C. Yu, Y. J. Jin, J. Wang, The Research of Missing Data Imputation Method: Based on Nearest Neighbor Imputation and Association Rules, *Journal of Statistics and Information*, Vol. 30, No. 1, pp. 35-40, January, 2015.

[17] X. L. Sun, Y. Guo, N. Li, X. X. Song, Missing Data Imputing Algorithm Based on Modified Neural Process, *Journal of University of Chinese Academy of Sciences*, Vol. 38, No. 2, pp. 280-287, March-April, 2021.

[18] F. Y. Bu, Z. K. Chen, Q. C. Zhang, Missing Value Imputation Algorithm Based on Clustering and Auto-encoder, *Computer Engineering and Applications*, Vol. 51, No. 18, pp. 13-17, September, 2015.

[19] H. Y. Chen, J. H. Du, W. N. Zhang, Monitoring Data Repairing Method Based on Deep Denoising Auto-encoder Network, *Systems Engineering and Electronics*, Vol. 40, No. 2, pp. 435-440, February, 2018.

[20] D. W. Xu, H. Peng, X. T. Shang, C. C. Wei, Y. F. Yang, Road Network Data Repair Based on Graph Autoencoder-generative Adversarial Network, *Journal of Transportation Systems Engineering and Information Technology*, Vol. 21, No. 6, pp. 33-41, December, 2021.

[21] X. G. Hu, H. G. Zhang, D. Z. Ma, R. Wang, Hierarchical Pressure Data Recovery for Pipeline Network via Generative Adversarial Networks, *IEEE Transactions on Automation Science and Engineering*, Vol. 19, No. 3, pp. 1960-1970, July, 2022.

[22] W. J. Zhang, G. Y. Xu, M. J. Li, S. Zhu, Missing Data Imputation Approach Based on Convolutional Neural Network, *Microelectronics & Computer*, Vol. 36, No. 3, pp. 48-52+57, March, 2019.

[23] S. X. Wang, L. Y. Guo, Q. Y. Zhao, A. C. Yang, Repair Method for Missed and Abnormal Wind Speed Data Based on Reference Sequence and Full Convolution Network, *Automation of Electric Power Systems*, Vol. 47, No. 9, pp. 129-136, March, 2023.

[24] Y. Hou, C. Y. Han, X. Zheng, Z. Y. Deng, Traffic Flow Data Repair Method Based on Spatial-temporal Fusion Graph Convolution, *Journal of Zhejiang University (Engineering Science)*, Vol. 56, No. 7, pp. 1394-1403, July, 2022.

[25] X. T. Chai, H. M. Gu, F. Li, H. Y. Duan, X. B. Hu, K. Lin, Deep learning for irregularly and regularly missing data reconstruction, *Scientific Reports*, Vol. 10, No. 1, Article No. 3302, February, 2020.

[26] Z. Y. Tang, Y. Q. Bao, H. Li, Group Sparsity-aware Convolutional Neural Network for Continuous Missing Data Recovery of Structural Health monitoring, *Structural Health Monitoring*, Vol. 20, No. 4, pp. 1738-1759, July, 2021.

[27] Q. S. Lin, X. M. Bao, C. X. Li, Deep learning based missing data recovery of non-stationary wind velocity, *Journal of Wind Engineering and Industrial Aerodynamics: The Journal of the International Association for Wind Engineering*, Vol. 224, Article No. 104962, May, 2022.

[28] B. C. Lu, D. M. Zhang, Q. Shu, Y. L. Li, Diagnosis and Repair of Traffic Fault Data Based on Spatio-Temporal Characteristics and Grey Residual, *Journal of Chongqing Jiaotong University (Natural Science)*, Vol. 39, No. 9, pp. 8-16, September, 2020.

[29] L. Dai, Y. Mei, S. G. Li, C. Qian, G. P. Wang, Traffic Flow Data Imputation Method Based on Symmetrical Residual U-Net, *Journal of Transportation Systems Engineering and Information*, Vol. 20, No. 5, pp. 93-99, October, 2020.

[30] Y. M. Shao, Y. Y. Gan, Y. T. Hou, Y. Zhong, Research on GA-RF Method Based on Traffic Flow Data Restoration, *Journal of Chongqing University of Technology (Natural Science)*, Vol. 35, No. 6, pp. 29-36, June, 2021.

[31] L. C. Li, X. Qu, J. Zhang, Y. G. Wang, H. C. Li, B. Ran, Missing Value Imputation Method for Heterogeneous Traffic Flow Data Based on Feature Fusion, *Journal of Southeast University (Natural Science Edition)*, Vol. 48, No. 5, pp. 972-978, September, 2018.

[32] X. X. Yu, B. P. Tang, W. Y. Wang, X. Y. Wu, B. Li, Repairing Deteriorated Data of Wind Turbines by Multi-head Attention Bi-directional Long Short Time Memory Networks under Complex Working Conditions, *Journal of Mechanical Engineering*, Vol. 59, No. 14, pp. 1-9, July, 2023.

[33] Z. X. Wang, J. J. Hu, B. Z. Liu, Recovery Method for Missing Measurement Data of Power Systems Based on

Long Short-Term Memory Networks, *Electric Power Construction*, Vol. 42, No. 5, pp. 1-8, May, 2021

[34] Y. Wang, F. T. Wang, J. L. Wang, B. Y. Song, Z. Shi, Missing Data Imputation Approach Based on Generalized Centroids Clustering Algorithm, *Journal of Chinese Computer Systems*, Vol. 38, No. 9, pp. 2017-2021, September, 2017.

[35] D. Liu, Y. Wang, C. Liu, K. Wang, X. Yuan, C. Yang, Blackout Missing Data Recovery in Industrial Time Series Based on Masked-Former Hierarchical Imputation Framework, *IEEE Transactions on Automation Science and Engineering*, Vol. 21, No. 2, pp. 1138-1150, April, 2024.

[36] T. Yang, Z. Z. He, D. Y. Zhang, H. B. Pen, H. Jiang, S. T. Cai, Y. D. Yuan, Y. P. Cai, Power Quality Data Loss Repair Algorithm Based on FSOM Neural Network, *Power System Technology*, Vol. 44, No. 5, pp. 1941-1949, May, 2020.

[37] C. Dai, S. P. Lu, C. Ma, S. Garg, M. Alrashoud, An Adaptive Rank-based Tensor Ring Completion Model for Intelligent Transportation Systems, *IEEE Transactions on Consumer Electronics*, Vol. 70, No. 1, pp. 2235-2243, February, 2024.

[38] C. Dai, Y. Zhang, Z. G. Zheng, A Nonlocal Similarity Learning-Based Tensor Completion Model with Its Application in Intelligent Transportation System, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 25, No. 3, pp. 3140-3151, March, 2024.

[39] S. P. Lu, P. Wang, W. H. Zhu, C. Dai, Y. Zhang, C. J. Liu, S. X. Dai, A Nonlocal Feature Self-Similarity Based Tensor Completion Method for Video Recovery, *Neurocomputing*, Vol. 580, Article No. 127513, May, 2024.

Biographies

Jun Tang is studying for a PhD at the College of Computer Science, Sichuan University. He is the head of Changhong's big data team. His main research areas are big data modelling and data governance.

Bing Guo is a professor and doctoral supervisor at the College of Computer Science, Sichuan University. He earned his PhD degree from University of Electronic Science and Technology of China in 2002. His main research interests are personal big data management and blockchain technologies.

Yan Shen is a professor and doctoral supervisor at the School of Control Engineering, Chengdu University of Information Technology. She obtained her PhD degree from University of Electronic Science and Technology of China in 2004. Her research primarily focuses on intelligent terminals and instruments.

Shengxin Dai obtained his PhD in computer science and technology from Sichuan University in 2017. Currently, he holds the position of a lecturer in computer science at Sichuan University. His ongoing research focuses on the verification of intelligent Cyber-Physical Systems, encompassing machine learning, formal methods, and embedded real-time systems.

Yuming Jiang is a professor in the College of Computer Science, Sichuan University in China. He received his MS and Ph.D. degrees in mechanical manufacture and automation from the Sichuan University in 1990 and 2001, respectively. His current research interests include intelligent manufacturing, big data and block chain.