Trend-Aware Data Imputation Based on Generative Adversarial Network for Time Series

Han Li, School of Information Science and Technology, North China University of Technology, China*  
Zhenxiang Liu, School of Information Science and Technology, North China University of Technology, China  
Jixiang Niu, School of Information Science and Technology, North China University of Technology, China  
Zhongguo Yang, School of Information Science and Technology, North China University of Technology, China  
Sikandar Ali, Department of Information Technology, The University of Haripur, China

ABSTRACT

To solve the problems of generative adversarial network (GAN)-based imputation method for time series, which are ignoring the implied trends in data and using multi-stage training that may lead to high training complexity, this article proposes a trend-aware data imputation method based on GAN (TrendGAN). It implements an end-to-end training using de-noising auto-encoder (DAE). It also uses bidirectional gated recurrent unit (Bi-GRU) in the generator model to consider the bi-directional characteristics and supplement the features lost by de-noising auto-encoder and improves the discriminator’s ability using Bi-GRU and hint vector. The authors conducted experiments on four real datasets. The results showed that all components introduced into the method contribute to enhancing the imputation accuracy, and the MSE values of TrendGAN are much lower than those of baseline methods when dealing with time series with random and continuous missing patterns. That is, TrendGAN is suitable for data imputation in complex scenarios with two missing patterns coexist, such as electric power and transportation.

KEYWORDS

bidirectional gated recurrent unit, data imputation, de-noising auto-encoder, generative adversarial network, time series

1. TREND-AWARE DATA IMPUTATION BASED ON GENERATIVE ADVERSARIAL NETWORK FOR TIME SERIES

The world is full of multi-variate time series data, and time series analysis has already played an important role in various fields, such as stock price prediction (Li & Yang, 2020), urban applications (Tabassum et al., 2021), geolocation (Chatzigeorgakidis et al., 2020), financial data modelling...
(Dogariu et al., 2022), satellite monitoring (Yuan et al., 2023), fault anomaly detection (Patel et al., 2022), and IoT device maintenance (Alghamdi et al., 2022). Time series, however, are often incomplete for equipment fault, transmission error, human factor, and for other reasons, which affects the effectiveness of data analysis.

Traditional methods of data imputation mainly fall into two categories: deletion-based method and filling-based method. The deletion-based method creates the illusion of no missing values by deleting missing samples, which can ensure the integrity of the remaining data, but it causes a decrease in the scale of samples and transforms the deleted samples from partial missing state to complete missing state. Therefore, it is not applicable to time series with continuous changing trend (Xu et al., 2020). The filling-based method fills the missing data by generating new values, and it can be further subdivided into statistical-based method and machine learning-based method. There are many commonly used statistics-based methods, such as mean imputation (Wolbers et al., 2022), last observation carried forward (Sampoornam et al., 2022), median imputation (Hadeed et al., 2020), plural imputation (Memon et al., 2022), random imputation (Guillaume & Wilfried, 2018), next observation carried backward (Wu et al., 2022), Lagrange imputation (Essanhaji & Errachid, 2022), and so on. Meanwhile, the main technologies used in the machine learning-based methods include clustering (Lashmar et al., 2021), linear regression (Vance et al., 2022), matrix decomposition (Feng et al., 2023), correlation analysis (Zhang et al., 2021), and multiple imputation (Aleryani et al., 2022). These methods mainly focus on the processing of missing values of non-time series.

Time series is a chronologically arranged sequence of numerical data points (Ren et al., 2021) and has seen extensive applications in various domains of our daily lives, especially in industrial scenarios. Since time series is generally generated by end-users, edge devices, and different wearable devices, it is more inevitable for time series to suffer missing values.

In recent years, neural networks have proved successful in multiple application domains, and plenty of data imputation models for time series have been designed based on neural networks. Che et al. (2018) investigated the approach to predicating time series based on recurrent neural networks. The method can capture the long-term temporal dependency and accurately predict missing data in time series. However, the imputation accuracy of the method may decrease, or the method may fail altogether if the correlation between missing attributes is unclear. William and Andrew (2018) proposed an auto-encoder-based time series data generation method, which should be trained on a complete time series, and a portion of the temporal data must be discarded manually to generate better data. That is, sufficient data preparation should precede training. Smith et al. (2010) filled the values of missing data by an ensemble learning method, which requires several iterations of the training process and consumes a considerable amount of memory. Thus, this method is ill-suited for larger datasets, and data scaling is required when it is used for processing data with certain scale. In brief, the above methods use the neural network to their advantage in terms of parallel processing ability and learning ability, while it is necessary to set a large number of parameters according to the actual application scenarios, and they also involve substantial preparation work for preliminary data.

Generative Adversarial Networks (GAN), a novel neural network model for estimating generative models via adversarial nets, was first proposed by Goodfellow et al. in 2014. As one of the most promising methods of complex unsupervised learning, it has already been applied to support data imputation. Yoon et al. (2018) proposed a GAN-based data imputation method (GAIN), in which Hint Matrix is used to mark part of the location information of missing values. The method can help the generator model to generate data that is more consistent with the distribution of the original data. It is, however, mainly proposed for data that is missing completely at random. Karras et al. (2018) proposed a new training methodology for generative adversarial networks, named Progressive Growing of GANs (PGGAN). Using a progressive approach to increase resolution gradually while training the generator and discriminator to improve the image quality, PGGAN performs well in generating high-resolution images. However, this method consumes a considerable amount of time and computation resources and is at risk of mode collapse. Dong et al. (2019) proposed a novel approach
based on GAN (SDAM), which can achieve both missing data imputation and data augmentation in one shot. This algorithm, however, is applicable only to datasets where distribution structure is fixed, and uneven or noisy data distribution may discourage its performance. Some methods focus on the imputation of time series data. Xu et al. (2019) proposed a data imputation method TabularGAN, in which GAN is utilized to generate tabular data similar to educational records or medical records. Using long short-term memory (LSTM) and attention mechanisms, this method can handle various types of features and data, including continuous, categorical, and time series features. However, this approach may struggle with complex table structures, and does not guarantee the quality and authenticity of the generated data when the missing values are present in multiple tables. Moreover, this method requires cleaning of the outliers and invalid data from the generated data, which entails some degree of human intervention and domain knowledge. To fill the missing values in time series, Li et al. (2020) proposed a novel generative adversarial network-based approach known as TimeGAN. This approach introduces a new pre-training method for the generator network that uses a truncated back-propagation-through-time (TBPTT) algorithm. However, this method may be susceptible to interference between patterns, which makes realistic data generation unattainable. Furthermore, data pre-training and feature engineering are required in the early stages to provide effective input data for GANs. Shang et al. (2017) proposed a multi-modal data-oriented data imputation method based on GAN, which can learn the common features of multi-modal data and fill in the missing data for a particular modal. The method improves considerably in processing multi-variate data, but it requires a multi-stage training which increases the risk of training failure. Considering the strong independence of solar data, a data imputation method (SolarGAN) was proposed for multi-variate solar data by W. Zhang et al. (2021). The method extends the input of Wasserstein GAN (WGAN) (Luo, Y. et al. 2018) with a combination of random noise and original data, which can retain part of the features of solar data, and process relatively independent data more effectively. However, the method only considers the unidirectional features of time series, and ignores the bidirectional features of time series. In a nutshell, GAN-based data imputation methods for time series can achieve good results without preliminary data preparation and are applicable to multi-variate time series. However, these GAN-based data imputation methods have the following limitations. Firstly, the implied trends (Ren et al., 2021) are not given full consideration, which may prevent the model from learning more correct data distribution. Secondly, using multi-stage training will increase the number of parameters and additional manual intervention. This approach may, to some degree, increase the training time and reduce the accuracy of data imputation. Hence, this work investigates a trend-aware data imputation method based on generative adversarial network for time series (TrendGAN), in which de-noising auto-encoder (DAE) and Bi-GRU are introduced to improve the GAN model for the purpose of considering the bidirectional features of time series, replacing a multi-stage training with an end-to-end training, and finally promoting the accuracy of data imputation.

The main contributions of this paper are summarized as follows:

- The generator model realizes an end-to-end training through de-noising auto-encoder-based dimensionality reduction, eliminating the need for multiple optimizations of input data and reducing errors caused by human factors.
- Bi-GRU is introduced to both generator and discriminator to extract the bidirectional features of time series, and it can compensate for the loss of features of time series during dimensionality reduction of DAE.
- Additional information is appended to the discriminator model to make the discriminator model focus on the imputation quality of specific locations and learn the real data distribution.
- Experiments on four real-world datasets show that the proposed method achieves state-of-the-art imputation accuracy in both random and continuous missing pattern.
The remainder of the paper is organized as follows: Section 2 presents the methodology of the proposed imputation method, which includes the problem statement and the details of TrendGAN. Section 3 describes the experimental setup and discusses the main results. Finally, the conclusion is reached in Section 4.

2 METHODOLOGY

2.1 Problem Statement

**Time step:** The set of time steps \( T = \{t_0, t_1, \ldots, t_{n-1}\} \) represent a series of time steps arranged in chronological order, where \( t_i \ (i \in \{0, 1, \ldots, n-1\}) \) denotes the \( i \)-th time step.

**Time series:** Time series refers to the object of imputation. It is represented as a matrix with \( d \) rows and \( n \) columns \( X = (X_0, X_1, \ldots, X_{n-1}) \in \mathbb{R}^{d \times n} \), where \( n \) represents the time step, \( d \) represents the dimension of attributes. \( X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,d}) \) denotes all the attributes at time step \( t_i \), and \( x_{i,j} \ (j \in \{1, 2, \ldots, d\}) \) denotes the \( j \)-th attribute at time step \( t_i \).

**Random noise:** Random noise is a matrix comprising of \( d \) rows and \( n \) columns, which is represented as \( Z = (Z_0, Z_1, \ldots, Z_{n-1}) \in \mathbb{R}^{d \times n} \), where \( Z_i = (z_{i,1}, z_{i,2}, \ldots, z_{i,d}) \) denotes all the random noise at time step \( t_i \), and \( z_{i,j} \ (j \in \{1, 2, \ldots, d\}) \) is a random variable in the interval \([0, 1]\).

**Mask matrix:** Mask matrix records the location information of the missing values in a dataset. It is a matrix comprising of \( d \) rows and \( n \) columns, which is denoted as \( M = (M_0, M_1, \ldots, M_{n-1}) \in \mathbb{R}^{d \times n} \), where \( M_i = (m_{i,1}, m_{i,2}, \ldots, m_{i,d}) \) represents the missing information of a dataset at time step \( t_i \), and \( m_{i,j} \ (j \in \{1, 2, \ldots, d\}) \) denotes whether the value of the \( j \)-th attribute is lost at time step \( t_i \). The value of \( m_{i,j} \) is 0 or 1, in which \( m_{i,j} = 0 \) indicates that the value of the \( j \)-th attribute is missing, while \( m_{i,j} = 1 \) is the opposite. In addition, the missing rate of time series can be controlled by adjusting \( M \).

**Time Series Imputation:** Based on the above definition, time series with missing data can be represented as a dataset \( X \) and its corresponding \( T \) and \( M \), and data imputation for time series is defined as the process of obtaining a dataset \( \bar{X} \) without missing data by filling the missing data in \( X \) with the generated data \( G \left( X, (1 - M) \odot Z \right) \) from the generator model.

2.2 Method Overview

Most time series have an inherent characteristic of trend (hereinafter referred to as trend), which is the change pattern in data. That is to say, time series are often accompanied by a certain trend or even a combination of various trends. In the task of data imputation for time series, if the trend hidden in the time series is correctly identified, this will allow the imputation model to be more consistent with the real data and thus the accuracy of imputation will be improved. In this paper, forward data refers to the data in front of the missing data, while backward data is the opposite. Since forward data directly affects the value of the missing data, most of the current data imputation methods construct the imputation model based on forward data. However, trend is always so continuous that backward data can also reflect trend and help to build a model that is more consistent with the real data distribution. Therefore, this paper comprehensively considers the impact of both forward data and backward data on the missing data and proposes a trend-aware data imputation method based on GAN for time series, attempting to enhance the imputation accuracy.
The proposed method, as illustrated in Figure 1, mainly comprises three parts including generator model, discriminator model, and imputation.

Below is a detailed description of TrendGAN.

**Generator model:** The generator model is responsible for generating new data. To avoid high training cost and loss of key information that arise from a multi-stage training, TrendGAN uses the compression and dimensionality reduction capability of de-noising auto-encoder (DAE) to achieve an end-to-end training. Since Bi-GRU can extract bidirectional characteristics of time series data, it is used after DAE to compensate for the potential information loss during the dimensionality reduction. In addition, the input of the generator model is extended to maintain the original data features to the greatest extent without destroying the randomness required by the training of generator model. The expanded input includes the incomplete time series, mask matrix, and random matrix, in which the mask matrix is used to mark the location information of missing values in the time series.

**Discriminator model:** The discriminator model is responsible for identifying whether the generated data are real or fake. Since a weak discriminator model can make the generator mode fail to converge, TrendGAN uses Bi-GRU and fully connected layers to enhance the ability of the discriminator model to deal with data imputation for time series. In addition, the input of discriminator model is also extended to make full use of the correct part of the generated data and enhance the attention to the imputation accuracy of specific locations.

**Imputation:** Imputation is responsible for filling the missing values to construct a complete dataset. In our work, imputation is defined as the process of replacing the missing values recorded by $M$ with a set of new values generated by the generator model.
2.3 Generator Model

An analysis of the well-accepted GAN-based data imputation methods for time series shows that the main problems of the generator model are as follows. Firstly, only the forward data is considered and the impact of backward data on the model is ignored, regardless of which the generator model uses, gate recurrent unit (GRU) or LSTM. Secondly, most methods rely on a multi-stage training to build the generator model, which increases the number of additional training parameters and manual intervention, and, to a certain extent, leads to the increase of training time and the reduction of imputation accuracy.

To solve the first problem, considering the bidirectional characteristics, Bi-GRU instead of GRU or LSTM is used, which can effectively introduce the backward data that reflects the trend hidden in time series. Additionally, bidirectional long short-term memory (Bi-LSTM) can also extract bidirectional features from time series. However, since the computational complexity of Bi-LSTM is higher than that of Bi-GRU, Bi-GRU is more suitable for this task.

To solve the second problem, de-noising auto-encoder is introduced into the generator model. Firstly, the original data is transformed into a feature vector with low dimensionality, and then decoder is used to reconstruct the low-dimensional feature expression vector into complete timing series. That is, a multi-stage training of the generator model is replaced by an end-to-end training. Part of the inherent features of time series, however, may be lost in the dimensionality reduction process of de-noising auto-encoder. Thus, Bi-GRU is used to enhance the output of the decoder, which can compensate the impact of feature loss. The structure of the generator model designed in this paper is illustrated in Figure 2.

As shown in Figure 2, the input of the generator model consists of time series X, mask matrix M, and random noise Z, where X is the original time series, M is utilized to denote the positions of all missing values in X, and M changes dynamically depending on X. Z contains the random noise used for data imputation. That is, assuming the input of the generator model is represented as a matrix with d rows and n columns \( \tilde{X} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n) \in \mathbb{R}^{d \times n} \), where \( \tilde{x}_i = (\tilde{x}_{i1}, \tilde{x}_{i2}, \ldots, \tilde{x}_{id}) \), \( \tilde{X} \) is calculated as Formula 1.
\[ \hat{X} = M \odot X \] (1)

Inside the generator model, the bidirectional features of the input are initially extracted by Bi-GRU and are passed to the de-noising auto-encoder. Then, new data are generated by the dimensionality reduction and reconstruction of de-noising auto-encoder. At last, Bi-GRU is applied to the new generated data to enhance the characteristics hidden in time series.

The output of the generator model \( \hat{X} \) is also a matrix with \( d \) rows and \( n \) columns, which can be represented as Formula 2.

\[ \hat{X} = G\left( \hat{X}, \left(1 - M\right) \odot Z \right) \] (2)

Since de-noising auto-encoder is a part of the generator model, the loss function should not only consider how to cheat the discriminator model, but also consider how to reconstruct the data more accurately. Thus, the generator’s loss function is defined as Formula 3, where the input of the discriminator model is \( \hat{X} \), and \( \tau \) is a hyperparameter.

\[ G_{loss} = -D\left( \hat{X} \right) + \tau \mid \hat{X} - X \mid_2 \] (3)

### 2.4 Discriminator Model

By analyzing the well-accepted GAN-based data imputation methods for time series, it can be found that the main problems of the discriminator model are as follows. Firstly, the discriminator model does not consider the backward data. Secondly, if the new data created by the generator model is directly used as the input of the discriminator model, the correct part of the generated data will be ignored by the discriminator mode and affects the training of the generator model indirectly.

To solve the first problem, bidirectional gate recurrent unit and fully-connected layer neural network (FC-NN) are used in the discriminator model. Bi-GRU is employed to make full use of the bidirectional characteristics of time series. FC-NN can thus increase the number of trainable parameters of the discriminator model and make the discriminator model more robust.

To solve the second problem, the input of the discriminator model is extended from the generated data to the combination of the generated data and the original data. Extending the input allows the discriminator model to judge whether part of the generated data is true or false, which will no longer ignore the contribution of the correct part of the generated data. Furthermore, a hint message is added to the discriminator model, which improves the quality of the data distribution learned by the generator model by making the discriminator model focus on specific parts of time series. The structure of the discriminator model designed in this paper is illustrated in Figure 3.

As shown in Figure 3, the input of the discriminator model is composed of dataset \( X \), dataset \( \hat{X} \), and mask matrix \( M \), where \( X \) is the original time series and \( \hat{X} \) is output of the generator model. In the discriminator mode, \( H \) contains the locations of \( X \) that are more important for improving the quality of data imputation. \( \hat{X} \) is a matrix with \( d \) rows and \( n \) columns \( \hat{X} = \left( \hat{X}_1, \hat{X}_2, \ldots, \hat{X}_n \right) \in \mathbb{R}^{d \times n} \), where \( \hat{X}_i = \left( \hat{x}_{i,1}, \hat{x}_{i,2}, \ldots, \hat{x}_{i,d} \right) \), and \( \hat{X} \) is constructed by combing \( X \), \( \hat{X} \), and Mask matix \( M \) according to Formula 4.

\[ \hat{X} = M \odot X + \left(1 - M\right) \odot \hat{X} \] (4)
Inside the discriminator model, the bidirectional features are firstly obtained by Bi-GRU from the input data and are passed to the fully connected layer (FC). Then the true probability $P(\text{real})$ of each generated data can be approximated with the multi-layer network of FC.

The hint message is also a matrix with $d$ rows and $n$ columns $H = [H_1, H_2, \ldots, H_{d \times n}] \in \mathbb{R}^{d \times n}$, where $H_i = (h_{i,1}, h_{i,2}, \ldots, h_{i,d})$, and $h_{i,j}$ ($j \in 1, 2, \ldots, d$) is a random variable. The value of $H$ is related to $M$ and another matrix $B$ with $d$ rows and $n$ columns. Matrix $B$ is used to describe the distribution of the hint message. The value of $b_{i,j}$ is 0 or 1, in which $b_{i,j} = 1$ indicates that there is a hint message in the corresponding location, and $b_{i,j} = 0$ is the opposite. The calculation of $H$ is shown in Formula 5.

$$H = B \odot M + 0.5(1 - B) \quad (5)$$

According to Formula 5, the value of $h_{i,j}$ can be 0, 0.5 or 1, in which $h_{i,j} = 1$ represents that the value in the corresponding position is real, $h_{i,j} = 0$ represents that the value in the corresponding position is generated by the generated model, and $h_{i,j} = 0.5$ represents whether the value of the corresponding position is real or not is unknown.

Moreover, by extending the input and introducing a hint message, Formula 6 presents the loss function of the discriminator model.

$$D_{\text{loss}} = -\left( D(X, H) - D(\hat{X}, H) \right) \quad (6)$$

### 2.5 Imputation

TrendGAN is used to construct a generator model capable of reflecting the true distribution of time series. Inside the generator model, time series $X$ is firstly mapped to a low-dimensional vector $\tilde{Z}$,
and then $\bar{Z}$ is used to reconstruct a complete time series $\bar{X}$ whose distribution should be made consistent with that of $X$ as much as possible. Finally, $\bar{X}$ is applied for impute the missing values in time series $X$ before the complete time series $X_{\text{imputed}}$ is obtained. The calculation method of $X_{\text{imputed}}$ is shown in Formula 7.

$$X_{\text{imputed}} = M \odot X + (1 - M) \odot \bar{X}$$

(7)

3 EVALUATION

3.1 Experimental Setting

Datasets: Experiments have been performed on four real-world datasets to verify the feasibility of our method. The details of these datasets are as follows:

Dataset 1: Electricity France is a dataset from the Kaggle website (Georges et al., 2021), and the dataset contains a total of 1,442 entries covering data on French households with daily electricity consumption data from December 2006 to November 2010. The dataset includes 6 types of attributes, including active power, reactive power, voltage, active power consumed in the kitchen, active power consumed in the laundry room, and active power consumed in heating, ventilation, and air conditioning (HVAC), etc.

Dataset 2: Spam is a dataset which is from the UCI machine learning repository (Lichman, 2013). It relates to 57 email feature attributes, with a total of 4601 entries. The dataset includes 6 types of attributes, such as word_frequency and types of spam. The dataset is collected on the daily basis and has been extensively studied and validated by numerous studies.

Dataset 3: Weather Madrid is a dataset from the Kaggle website and is provided by the Weather Company, LLC (LLC, 2016). It collects the weather data of Madrid Barajas Airport from January 1997 to December 2015, with a total of 6812 entries. It includes 23 climate attributes, such as temperature, humidity, and visibility kilometers, etc., and is collected on a daily basis. The dataset has passed rigorous quality checks and has been certified to be highly reliable and accurate for use in climate analysis and research.

Dataset 4: Water Quality is a dataset available from the Kaggle website (Aditya, 2021), which collects water quality data from various water sources, such as rivers, lakes, and groundwater, with a total of 3276 entries. The dataset includes 9 water quality attributes, such as pH, hardness, conductivity, and organic mineral content, as well as a target variable indicating whether the water is safe for human consumption or not. The data is collected on the daily basis and has been verified for accuracy and completeness, making it a reliable resource for studying related issues.

Missing rate (MR) is defined as the ratio of the number of missing data to the number of the whole data. To assess the effectiveness of TrendGAN thoroughly, experiments must be performed on datasets that have various missing rates. Thus, a random method is used to set the missing rate on four datasets, and the missing rate is set to 20%, 50%, 70%, and 90%. In addition, 20% of the dataset was classified as the testing set, 10% as the verification set, and 70% as the training set.

Baselines: To validate the performance of TrendGAN, the following methods are selected for comparison. These comparison methods are detailed as follows:

Mean Imputation (Mean): It is a statistics-based imputation method. The mean value is directly used to fill in the missing value.

KNN Imputation (KNN): KNN is a classical machine learning method. KNN-based imputation refers to filling in the missing values with the mean value of the k nearest neighbors.
Matrix Factorization Imputation (MF): As a machine learning-based imputation method, MF treats the original dataset as a matrix and uses the matrix decomposition method to obtain two low-rank matrices. Then the product of these two low-rank matrices is used to fill in the missing values.

Multiple Imputation using Chained Equations (MICE): MICE as a machine learning-based imputation method uses an iterative auto-regressive method to fill in the missing values. Its time consumption may go up accordingly as the number of iterations increases.

GAIN: GAIN is a GAN-based data imputation method. Some extra information in the form of hint vectors is provided to the discriminator model, which helps the discriminator model to focus on specific components and improve the quality of imputation for those components.

SolarGAN: As a GAN-based imputation method, SolarGAN changes the input of WGAN from pure random noise to a combination of real samples and random noise and can be used to fill in the missing value of multi-variate data.

Evaluation Metrics: The Mean Square Error (MSE) between the original dataset without missing data and the imputed dataset is used to evaluate the performance of TrendGAN. The calculation of MSE is shown in Formula 8, where $x_{t_i}$ denotes the original dataset at time step $t_i$, $x_{imputed_{t_i}}$ denotes the imputed dataset at time step $t_i$ and $k$ is total number of time steps. Furthermore, the higher the MSE, the worse the imputation accuracy.

$$MSE = \frac{1}{K} \sum_{i=1}^{k} (x_{t_i} - x_{imputed_{t_i}})^2$$ (8)

Environments: The experiment is conducted on a XenServer virtual machine with the following configuration: Ubuntu 10.3.0, one Intel Xeon Platinum 8163, 110 GB RAM, 1TB hard disk. Additionally, TrendGAN and all baselines are implemented in Python and PyTorch 1.10.0.

Hyperparameter Settings: There are 5 hyper-parameters including $\lambda$, $\tau$, learning_rate, batch_size and epoch. Table 1 lists the setting of the hyper-parameters in the experiments. $\lambda$ is the parameter that controls the proportion of hint message in the discriminator model; $\tau$ is the parameters used to balance the proportion of GAN and de-noising auto-encoder in the loss function. batch_size and learning_rate are the hyperparameters used to train the deep learning model. epoch is the parameter to control the number of iterations.

3.2 Experiments

Ablation Experiment of TrendGAN: To verify the contribution of each component in TrendGAN, the ablation experiment of TrendGAN performed by sequentially excluding each component is carried out on four datasets, and the results of the ablation experiment is given in Table 2.

As shown in Table 2, TrendGAN outperforms the rest methods in all components in the ablation experiment, and a detailed analysis is given below.

<table>
<thead>
<tr>
<th>Hyper-parameter</th>
<th>DataSet1</th>
<th>DataSet2</th>
<th>DataSet3</th>
<th>DataSet4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda$</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
</tr>
<tr>
<td>$\tau$</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>learning_rate</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>batch_size</td>
<td>64</td>
<td>64</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>epoch</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
</tr>
</tbody>
</table>
Firstly, when the de-noising auto-encoder is removed, the MSE value is increased by 34.3% on average with a minimum rise of 19.74%. That is, the de-noising auto-encoder turning a multi-stage training into an end-to-end training can reduce the number of errors caused by human intervention and additional training parameters and improve the performance of the generator model significantly.

Secondly, with the first Bi-GRU layer, the second Bi-GRU layer and both the first and the second Bi-GRU layers in the generator model removed, the MSE value is increased by 30.7%, 21.6%, and 20.4%, respectively on average, and when the Bi-GRU layer is removed from the discriminator model, the MSE value increases 24.0% on average. That is, considering the backward data can provide more information, thus the imputation accuracy of time series is improved with hidden trends. In addition to increase the MSE value by removing the second layer of Bi-GRU from the generator model shows that it is effective to use Bi-GRU to compensate for the possible data loss caused by de-noising auto-encoder.

Thirdly, when Hint is removed from the input of the discriminator model, the MSE value goes up 30.8% on average, which suggests that focusing on the internal features of the generated data can improve the TrendGAN performance. The inclusion of Hint in the discriminator model, however, requires more pre-processing of the inputs.

**Experiment on Imputation Accuracy with Different Missing Rates:** Table 3 to Table 6 show the imputation accuracy, respectively, in terms of MSE for the baseline methods and TrendGAN on four datasets with random missing mode.

The above experimental results show TrendGAN outperforms other baseline methods, and a detailed analysis is as follows.

Firstly, the rise in the missing rate is accompanied by an increase in the MSE values of TrendGAN and all baseline methods. This is to suggest that the missing rate impacts the imputation accuracy: The higher the missing rate, the lower the imputation accuracy.

### Table 2. Imputation accuracy of ablation experiment in terms of MSE Score on four datasets

<table>
<thead>
<tr>
<th>Method</th>
<th>Dataset 1</th>
<th>Dataset 2</th>
<th>Dataset 3</th>
<th>Dataset 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TrendGAN</td>
<td>TrendGAN</td>
<td>G w/o first-</td>
<td>G w/o second-</td>
</tr>
<tr>
<td></td>
<td>w/o DAE</td>
<td>w/o first-BiGRU</td>
<td>BiGRU</td>
<td>BiGRU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>G w/o second-</td>
<td>D w/o</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>BiGRU</td>
<td>BiGRU</td>
</tr>
<tr>
<td>Dataset 1</td>
<td>0.0113</td>
<td>0.01169</td>
<td>0.0200</td>
<td>0.0165</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(33.1%)</td>
<td>(43.5%)</td>
<td>(31.5%)</td>
</tr>
<tr>
<td>Dataset 2</td>
<td>0.0020</td>
<td>0.0032</td>
<td>0.0029</td>
<td>0.0027</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(37.5%)</td>
<td>(31.0%)</td>
<td>(25.9%)</td>
</tr>
<tr>
<td>Dataset 3</td>
<td>0.0138</td>
<td>0.0260</td>
<td>0.0234</td>
<td>0.0182</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(46.9%)</td>
<td>(41.0%)</td>
<td>(24.2%)</td>
</tr>
<tr>
<td>Dataset 4</td>
<td>0.0317</td>
<td>0.0395</td>
<td>0.0342</td>
<td>0.0333</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(19.7%)</td>
<td>(7.3%)</td>
<td>(4.8%)</td>
</tr>
</tbody>
</table>

### Table 3. Imputation accuracy in terms of MSE with different missing rates, on dataset 1

<table>
<thead>
<tr>
<th>Method</th>
<th>MR</th>
<th>Mean</th>
<th>KNN</th>
<th>MF</th>
<th>MICE</th>
<th>GAIN</th>
<th>SolarGAN</th>
<th>TrendGAN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>20%</td>
<td>0.917</td>
<td>1.404</td>
<td>0.628</td>
<td>0.631</td>
<td>0.021</td>
<td>0.974</td>
<td>0.011</td>
</tr>
<tr>
<td></td>
<td>50%</td>
<td>0.957</td>
<td>1.039</td>
<td>0.791</td>
<td>0.931</td>
<td>0.031</td>
<td>1.038</td>
<td>0.018</td>
</tr>
<tr>
<td></td>
<td>70%</td>
<td>0.997</td>
<td>1.051</td>
<td>0.896</td>
<td>1.132</td>
<td>0.055</td>
<td>1.019</td>
<td>0.021</td>
</tr>
<tr>
<td></td>
<td>90%</td>
<td>1.000</td>
<td>1.063</td>
<td>0.959</td>
<td>1.218</td>
<td>0.217</td>
<td>1.044</td>
<td>0.025</td>
</tr>
</tbody>
</table>
Secondly, TrendGAN is smaller than the baselines in MSE value regardless of the value of the missing rate, which shows that the proposed method can optimize the imputation accuracy of time series. Additionally, the MSE values of the GAN-based imputation methods (GAIN, SolarGAN, TrendGAN) are at least 30.0% and at most 99.2% lower than those of the statistical-based imputation method (Mean) and the machine learning-based imputation methods (KNN, MF, MICE). This is because GAN as a method better suited for complex unsupervised learning can generate new data, which is more consistent with the distribution of the original data, especially for multivariate time series with implicit trend information.

Thirdly, among the three GAN-based imputation methods, SolarGAN performs far worse than GAIN and TrendGAN in imputation accuracy. That is because either the generator model or the discriminator model of SolarGAN is composed of only GRU and a neural network model, which ignore the influence of backward data on imputation accuracy, and finally results in a lower imputation accuracy. Furthermore, the end-to-end training of Trend GAN can avoid the possible errors induced by the multi-stage training of SolarGAN.

Fourthly, although GAIN achieves the best imputation accuracy in all baseline methods, it is still lower than that of TrendGAN. To be more specific, on average, TrendGAN is about 58.1% lower than GAIN in MSE values, and the gap continues to widen as the missing rate goes up. Three factors account for this result. Firstly, the bidirectional features of time series are not considered in GAIN. The second reason is that the multi-stage training increases the risk of errors. The last
factor is that the input of the discriminator model in GAIN is only composed of a random noise, and the possible noise deviation may cause a decline in imputation accuracy.

Experiment on Imputation Accuracy with Continuous Missing Pattern: In actual production, data is not always missing at random. Datasets with continuous missing data are also common. To assess the imputation accuracy ability more comprehensively, the imputation accuracy of different data imputation methods in terms of MSE on Dataset 1 with continuous missing mode is given in Table 7. In the experiment, the window size, specifically the sliding window size of the missing values distributed over the time series, is set from 1 to 15, the position of windows is randomly distributed in the dataset, and the missing rate is set to 20%.

Below is a detailed analysis of the experimental results.

Firstly, the imputation accuracy of each method drops as the window size keeps increasing, but TrendGAN still stands out among all methods in MSE value. Secondly, the statistical-based method and the machine learning-based method are outperformed by other methods in imputation accuracy, while the GAN-based methods perform best in imputation accuracy. This is because statistic theory underlies the first two categories of imputation methods, and thus they are better suited for datasets where the missing pattern is random. For dataset with continuous missing data, these methods employ the same processing steps as those used for handling datasets with randomly missing data, which ignores the implicit correlation between data. Different from the above methods, GAN-based imputation methods generate new data according to the data distribution learned from the dataset with continuous missing data. That is, in these methods data is not considered independent, and they use the correlation between data to their full advantage. Therefore, it can be concluded that the GAN-based methods are more reasonable, and the more hidden information in time series, the better imputation accuracy of GAN-based methods.

Table 7. Imputation accuracy in terms of MSE score with different sliding window size on dataset 1

<table>
<thead>
<tr>
<th>Method Window Size</th>
<th>Mean</th>
<th>KNN</th>
<th>MF</th>
<th>MICE</th>
<th>GAIN</th>
<th>SolarGAN</th>
<th>TrendGAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.9178</td>
<td>1.4044</td>
<td>0.6285</td>
<td>0.6316</td>
<td>0.0212</td>
<td>0.9740</td>
<td>0.0113</td>
</tr>
<tr>
<td>2</td>
<td>0.9914</td>
<td>1.1607</td>
<td>0.6530</td>
<td>0.6562</td>
<td>0.0130</td>
<td>0.9425</td>
<td>0.0122</td>
</tr>
<tr>
<td>3</td>
<td>0.8856</td>
<td>1.0093</td>
<td>0.6186</td>
<td>0.6370</td>
<td>0.0136</td>
<td>0.9201</td>
<td>0.0132</td>
</tr>
<tr>
<td>4</td>
<td>0.8092</td>
<td>1.4045</td>
<td>0.5375</td>
<td>0.7392</td>
<td>0.0267</td>
<td>1.0416</td>
<td>0.0149</td>
</tr>
<tr>
<td>5</td>
<td>0.9284</td>
<td>1.1455</td>
<td>0.6144</td>
<td>0.6383</td>
<td>0.0143</td>
<td>1.1767</td>
<td>0.0141</td>
</tr>
<tr>
<td>6</td>
<td>1.0051</td>
<td>1.0983</td>
<td>0.6765</td>
<td>0.7102</td>
<td>0.0164</td>
<td>1.0993</td>
<td>0.0145</td>
</tr>
<tr>
<td>7</td>
<td>1.1209</td>
<td>0.8427</td>
<td>0.7733</td>
<td>0.7743</td>
<td>0.0158</td>
<td>0.9232</td>
<td>0.0154</td>
</tr>
<tr>
<td>8</td>
<td>0.8589</td>
<td>0.9898</td>
<td>0.5845</td>
<td>0.5835</td>
<td>0.0194</td>
<td>0.9055</td>
<td>0.0177</td>
</tr>
<tr>
<td>9</td>
<td>1.0342</td>
<td>0.8820</td>
<td>0.6992</td>
<td>0.6794</td>
<td>0.0278</td>
<td>1.0893</td>
<td>0.0202</td>
</tr>
<tr>
<td>10</td>
<td>0.8991</td>
<td>0.9875</td>
<td>0.6098</td>
<td>0.6277</td>
<td>0.0221</td>
<td>1.0102</td>
<td>0.0157</td>
</tr>
<tr>
<td>11</td>
<td>0.6233</td>
<td>1.1521</td>
<td>0.5825</td>
<td>0.6233</td>
<td>0.0191</td>
<td>0.9742</td>
<td>0.0189</td>
</tr>
<tr>
<td>12</td>
<td>1.0089</td>
<td>0.6233</td>
<td>0.7362</td>
<td>0.7642</td>
<td>0.0166</td>
<td>1.1854</td>
<td>0.0141</td>
</tr>
<tr>
<td>13</td>
<td>0.8531</td>
<td>0.9425</td>
<td>0.6152</td>
<td>0.6044</td>
<td>0.0171</td>
<td>1.2185</td>
<td>0.0130</td>
</tr>
<tr>
<td>14</td>
<td>0.9864</td>
<td>1.0028</td>
<td>0.7192</td>
<td>0.7068</td>
<td>0.0205</td>
<td>1.2108</td>
<td>0.0185</td>
</tr>
<tr>
<td>15</td>
<td>0.9055</td>
<td>0.8100</td>
<td>0.6207</td>
<td>0.6518</td>
<td>0.0242</td>
<td>0.8603</td>
<td>0.0189</td>
</tr>
</tbody>
</table>
Thirdly, the trend of the MSE score with different sizes of missing windows is shown in Figure 4.

As shown in Figure 4, although GAIN and TrendGAN stand out among all the methods in imputation accuracy, TrendGAN still outperforms GAIN, which is revealed more clearly in the locally zoomed images. That is, the MSE value of TrendGAN is about 27.3% lower than that of GAIN on average. Moreover, as the window size increases, the overall imputation accuracy of TrendGAN and GAIN drops 54% and 86% respectively, comparing the process from the minimum to maximum missing windows. That is, the imputation stability of TrendGAN is better than that of GAIN.

To sum up, the above experiments show that TrendGAN can deal with time series with randomly missing data or continuous missing data.

4 CONCLUSION

This paper investigates a trend-aware data imputation method based on GAN (TrendGAN) in order to overcome the main problems in GAN-based data imputation method. Firstly, to avoid possible errors caused by the multi-stage training whose complexity is quite high, de-noising auto-encoder is used to carry out an end-to-end training in the generator model. Secondly, both the forward data and the backward data are applied in GAN through Bi-GRU, which makes GAN comprehensively learn the data distribution of time series and compensate for the information loss caused by de-noising auto-encoder. Thirdly, the ability of the discriminator model is enhanced by providing more fine-grained information in its input. Experimental results on four real datasets show that all components of TrendGAN can help improve the accuracy of time series imputation. In addition, compared with the baseline methods, TrendGAN achieves the lowest MES value and the highest imputation accuracy when dealing with missing data in both random and continuous missing patterns. That is, TrendGAN
is suitable for promoting the quality of data in complex industry scenarios and can effectively improve the accuracy of further analysis and decision making. Furthermore, more comprehensive experiments will be conducted to analyze the impact of various missing patterns on imputation accuracy, and we will study more domain-targeted time series imputation methods.

DATA AVAILABILITY

Dataset 1, Dataset 2, Dataset 3, and Dataset 4 used to support the findings of this study are included within the article.

CONFLICTS OF INTEREST

The authors declare that there is no conflict of interest regarding the publication of this paper.

FUNDING STATEMENT

This work was supported by Beijing Municipal Natural Science Foundation (No. 4192020) and the Key-Area Research and Development Program of Guangzhou City (No. 202206030009).
REFERENCES


