Knowledge-Based Systems 283 (2024) 111215

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A time series continuous missing values imputation method based on generative adversarial networks[☆]

Yunsheng Wang [a], Xinghan Xu [b,*], Lei Hu [a], Jianchao Fan [a,c], Min Han [d,e]

[a] Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian, 116024, Liaoning, China
[b] Faculty of Infrastructure engineering, Dalian University of Technology, Dalian, 116024, Liaoning, China
[c] Department of Ocean Remote Sensing, National Marine Environmental Monitoring Center, Dalian, China
[d] Key Laboratory of Intelligent Control and Optimization for Industrial Equipment of Ministry of Education, Dalian University of Technology, Dalian, 116024, Liaoning, China
[e] Professional Technology Innovation Center of Distributed Control for Industrial Equipment of Liaoning Province, Dalian University of Technology, Dalian, 116024, Liaoning, China

ARTICLE INFO

Keywords:
Generative adversarial networks
Time series
Continuous missing
Data imputation
Adversarial learning

ABSTRACT

Generative adversarial networks (GANs) have been widely utilized in time series analysis and modeling, wherein generators and discriminators interact to generate realistic data. However, when addressing the challenge of imputing continuous missing values in time series, the generator struggles to learn meaningful features due to the loss of local information, and the discriminator's loss function exhibits significant deviations in determining the probability, making it difficult to effectively update model parameters during training. In response to these issues, this study presents a novel imputation model called cue Wasserstein generative adversarial network with gradient penalty (CWGAIN-GP). CWGAIN-GP incorporates the contextual cue information matrixs into the generator, enabling obtain and capture the potential data evolution features that hidden beyond the missing positions. This approach constrains the generator's generation results to be closer to the true values probability distribution. Meanwhile, it optimizes the generator and discriminator neural network structure and loss function computation strategy of the original GAN model, which not only improves the accuracy of continuous missing data interpolation, but also improves the training stability of the model. Finally, We used three imputation accuracies on two real-world datasets, including experiments on imputation accuracy for eight comparison models, ablation study, and experiments on the effect of consecutive missing lengths. The experimental results show that CWGAIN-GP achieves impressive performance in imputing continuous missing time series.

1. Introduction

Time series data plays a critical role in the real-world. It not only contains rich dynamic time information, but also has complex coupling relationships between its variables [1]. Extracting meaningful information from such data to construct high-precision prediction models holds great practical significance, providing theoretical basis for solving control problems in different fields [2]. However, the construction of accurate time series prediction models hinges upon the availability of complete and high-quality datasets [3]. Regrettably, missing data poses a common challenge in time series modeling, arising from factors such as equipment failures during data acquisition and transmission errors [4]. As the rate of missing data increases, the amount of valid information in the data decreases. Inability to build accurate models and inevitably large errors in the model prediction results will have a significant impact on the subsequent analysis and decision making [5].

The categorization of missing values in time series data generally encompasses three types: missing completely at random, missing at random, and not missing at random [6]. Moreover, the pattern of continuous missing does not strictly fall into the aforementioned three categories. In this pattern, all variables may exhibit regular or irregular missing values, and the duration of the continuous missing is unspecified. The occurrence of continuous missing values is prevalent in scientific research or industrial production scenarios. For instance, in fields like meteorology or water resources, sensor equipment often

[☆] This work was supported by the National Natural Science Foundation of China (62173063).

[*] Corresponding author.

E-mail addresses: wys222@mail.dlut.edu.cn (Y. Wang), xuxh2023@dlut.edu.cn (X. Xu), hl666888@mail.dlut.edu.cn (L. Hu), fjchao@dlut.edu.cn (J. Fan), minhan@dlut.edu.cn (M. Han).

https://doi.org/10.1016/j.knosys.2023.111215
Received 28 July 2023; Received in revised form 1 November 2023; Accepted 15 November 2023
Available online 17 November 2023

necessitates periodic maintenance, or certain equipment malfunctions unexpectedly, resulting in the inability to collect data for a specific period [7]. In wind or solar power generation scenarios, it is difficult to avoid varying degrees of missing data from the feedback due to weather changes and harsh environments. These data with missing data are not conducive to condition monitoring and fault diagnosis of power generation equipment [8]. The extent of missing data during such periods can be substantial, potentially impacting one or more parameters. These continuous missing values frequently coincide with extensive and long time series, leading to a notable loss of information and disruption in the data's distribution characteristics, as well as posing challenges concerning data availability, accuracy, and reliability. Prior to utilizing these valuable data, it is imperative to engage in missing data pre-processing [9].

Missing value handling during the pre-processing phase of data analysis has been a prominent research area. Three commonly employed methods for missing data pre-processing are the deletion method [10], no processing, and imputation method. While the deletion method simply removes observations with missing values, and the no processing approach disregards missing values altogether, imputation methods have garnered increased attention due to the growing demand for accurate data research outcomes [11]. Imputation methods aim to estimate or fill in the missing values based on available information, enabling the utilization of complete datasets for analysis. This ensures that the impact of missing data on subsequent analyses is minimized, and more accurate research results can be obtained.

Traditional statistical-based imputation methods, such as mean imputation and multiple imputation of chained equations (MICE), etc. [12], are relatively straightforward to implement. These methods exhibit good imputation accuracy when addressing missing data in linear and smooth time series. However, their accuracy tends to be compromised in complex time series with nonlinear characteristics, as they fail to account for the correlations and dependencies among different variables or data [13]. In recent years, with the surge of artificial intelligence, machine learning algorithms and other approaches have gained prominence in imputing missing data [14]. In the domain of machine learning, the concept of imputation can be categorized into two distinct models: discriminative models and generative models [15]. For example, Cao et al. [16] proposed the bidirectional recurrent imputation time series (BRITS) model, which estimates missing values based on the hidden states of a bidirectional recurrent neural network (BRNN). This model leverages the powerful learning capabilities of RNNs to address missing data imputation. Another notable approach is the self-attention imputation time series (SAITS) model proposed by Du et al. [17]. SAITS is based on the self-attention mechanism and is trained using a joint optimization method. It learns missing values through a weighted combination of two diagonally masked self-attention blocks. However, when confronted with vast amounts of time series data, discriminative models tend to exhibit slow convergence, posing a challenge in practical applications. As a result, generative models have gained popularity in the field of imputing missing values within multivariate time series [18].

Variational self-encoders (VAE) [19] and generative adversarial networks (GAN) [20] are increasingly used as mainstream generative models due to their good unsupervised properties [21]. For instance, Luo et al. [18] proposed an improved recurrent neural network (RNN) model as both the generator and discriminator in a generative adversarial network for imputing missing data in time series. Their approach enhances the capabilities of the GAN framework for time series imputation. Similarly, Yang et al. [22] developed an enhanced recursive imputation framework for time series based on GAN. Their method incorporates observations as additional inputs and leverages temporal and feature correlations among observations to reconstruct the time series data. In the domain of traffic prediction systems, Boquet et al. [23] applied VAE models to address the imputation task by utilizing the latest developments in variational inference. However, these models require the availability of a complete dataset for training, which is unrealistic in real-world scenarios because the original data itself is missing [24]. Thus Yoon et al. [25] proposed generative adversarial imputation nets (GAIN), Luo et al. [26] proposed end-to-end generative adversarial network (E2-GAN), Lai et al. [27] proposed an autoencoder-based multi-task learning mechanism (aeMTLDy), Ni et al. [28] proposed an improved generative adversarial network with multi head self-attention and bidirectional RNN (MBGAN), and Pan et al. [9] proposed an adaptive-learned median-filled deep autoencoder (AM-DAE) can avoid this problem. These methods utilize GAN, RNN or add an attention mechanism to improve the imputation accuracy.

It is worth noting that the majority of existing imputation models have not been specifically optimized to address the challenges associated with continuous missing data. Moreover, the experiments conducted in these models often focus on the ideal scenario of completely random missing data, which may lack realistic significance. For instance, the GAIN imputation method assumes a specific type of missing data, and when applied to imputation continuous missing data, the generated data distribution deviates significantly from the true data distribution. Additionally, the GAIN model, sharing the same generator and discriminator loss functions as the GAN model, is prone to encountering the problem of pattern collapse during the imputation process [29]. This issue arises from the flawed loss function and leads to the generation of data that lacks diversity and fails to accurately represent the underlying data patterns. These limitations highlight the need for further research and improvement in imputation models to specifically address the challenges posed by continuous missing data.

In order to address the limitations and structural flaws of the GAIN model in handling continuous missing time series data, this paper introduces a novel unsupervised approach called cue Wasserstein generative adversarial imputation network with gradient penalty (CWGAIN-GP). The CWGAIN-GP model is designed to overcome the shortcomings of existing approaches by incorporating a gradient penalty (GP) mechanism. This helps to address the problem of pattern collapse and encourages the generation of diverse and realistic imputed data. By leveraging the Wasserstein distance, the model strives to approximate the true data distribution more accurately and ensure the generated data aligns closely with the original data characteristics. Through the introduction of the CWGAIN-GP model, this paper aims to provide an improved solution for imputing continuous missing time series data.

The contributions of this paper can be summarized as follows:

1. We propose a novel generative adversarial imputation network that generates the imputation values in the continuous missing time series. The method constrains the range of imputed data generated by the generator and improves the generator generation accuracy by adding a cue information matrix to the generator.
2. We improve the structure of the generator and discriminator models in our proposed approach. Specifically, we eliminate the hint generator in GAIN. Additionally, we adopt the generator and discriminator loss function calculation strategy from Wasserstein GAN [29]. By incorporating these modifications, our proposed model addresses the issue of the loss function not reaching its minimum value during the process of imputing missing data.
3. We introduce the GP term to the discriminator of our proposed model. This addition restricts the range of the loss function, thereby preventing the discriminator parameters from diverging towards incorrect extremes that would lead to an increased loss function. Consequently, this enhancement strengthens the modeling capability of our proposed approach.

The rest of the paper is organized as follows. In Section 2, we briefly introduce some preliminaries about the GAIN model and WGAN. Our cue Wasserstein generative adversarial imputation network with gradient penalty model is proposed in Section 3. We evaluate the performance of our proposals in Section 4. Finally, Section 5 draws the conclusions and future work.

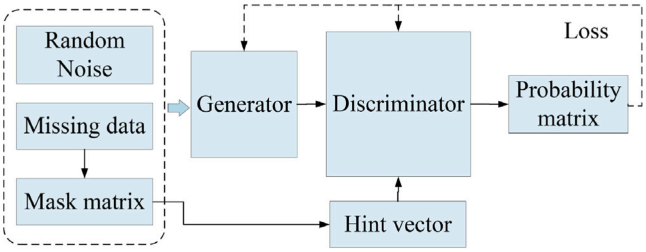


Fig. 1. Structure of GAIN model.

2. Preliminaries

We review few-shot the model principles and problems of imputation of missing values in time series based on GAN in this section, which are concentrated in our research work.

2.1. Generative adversarial imputation networks

In the past few years, there has been a surge of interest in GAN as a novel form of unsupervised learning [30]. GAN, which are neural network models, have demonstrated remarkable performance and great potential in various applications [31], such as image synthesis, image style transformation and image classification [32]. Leveraging their versatility and impressive capabilities, GAN models have also been utilized for imputing missing data in time series datasets.

Yoon et al. [25] introduced GAIN, which is built upon the original structure of GAN [33]. GAIN incorporates the principles of game theory and employs two neural networks, namely the generator and the discriminator [34], to iteratively update their parameters in a competitive manner. In the GAIN framework, the generator imputes missing values based on the distribution of observed missing data variables, while the discriminator assesses the facticity of the imputed results. The feedback from the discriminator is then used to train both the generator and the discriminator. To facilitate the discriminator in emphasizing the imputation quality of specific components, the GAIN model introduces cue vectors as additional information. These cue vectors are provided to the discriminator, allowing it to focus on improving the imputation quality for particular feature components. The GAIN model structure diagram is shown in Fig. 1.

Consider an array, denoted as $X = \{X_1, X_2, \dots, X_d\}$, in d-dimensional space. The d elements within X are assumed to be independent of each other, following a probability distribution represented by $P(x)$. Additionally, we have a mask matrix, $M = \{M_1, M_2, \dots, M_d\}$, associated with X. Whenever an element in X is zero or missing, the corresponding position in the mask matrix is denoted as “0”, while the elements in other positions are represented as “1”.

The basic structure of the generator is shown in Eq. (1):

$$Y = G(\hat{X}, M, (1-M) \odot N) \tag{1a}$$

$$\hat{X} = M \odot X + (1-M) \odot Y \tag{1b}$$

where Y is the imputed matrix, $\hat{X}$ is the final output matrix, X is the true sample matrix, N is the random noise, and $G(\cdot)$ is the generation process.

The GAIN model introduces a hint matrix H. The discriminator then inputs the real and generated data together with the hint vector, so that the probability value of each missing position can be represented.

The objective function can be expressed as

$$\min_G \max_D E_{\hat{X},M,H}[M^T \log D(\hat{X}, H) + (1-M)^T \log(1 - D(\hat{X}, H))] \tag{2}$$

where, $D(\cdot)$ is the discriminative process.

The loss function of the generator and discriminator can be showed as

$$L(a,b) = \sum_{i=1}^{d} [a_i \log(b_i) + (1-a_i) \log(1-b_i)] \tag{3}$$

where, $a = M$, $b = D(\hat{X}, H)$.

2.2. Wasserstein generative adversarial networks

In practical scenarios, when the original GAN model is engaged in a contest between the generator and the discriminator, an uneven match between the two parties, where one is strong and the other is weak, can result in oscillation and the failure of the objective function to converge. Consequently, the optimization algorithm fails to discover the Nash equilibrium, ultimately leading to pattern collapse in the model.

In order to address the aforementioned challenges, Martin et al. [29] proposed theoretical arguments and algorithmic modifications. Firstly, they incorporated a Lipschitz condition into the discriminator [35]. This constraint is showed as

$$|D(x) - D(y)| \le K|x-y| \tag{4}$$

where K is the Lipschitz constant, if the minimum value of K is obtained, x is the true data and y is the generated data.

Secondly, WGAN model introduces the Wasserstein distance [36] to measure the difference between the distribution of the generated data Y and the true data X. The Wasserstein distance formula can be expressed as

$$W(P_r, P_g) = \inf_{\gamma \sim \prod(P_r, P_g)} E_{(x,y)}[\|x-y\|] \tag{5}$$

where P_g and P_r are the distributions of the generated data samples and the true data samples, respectively, and γ is the joint distribution of the two. When the “complexity” of the transformation from P_g to P_r is minimized, it is $W(P_\gamma, P_g)$. However, since $W(P_\gamma, P_g)$ is difficult to solve, (5) can also be written as

$$W\left(P_r, P_g\right) = \frac{1}{K} \mathrm{Sup}_{\|f\|_L \le K^{x \sim p_r}} E_{x \sim p_r}[f(x)] - E_{x \sim p_g}[f(x)] \tag{6}$$

where $f(\cdot)$ is a function that satisfies the Lipschitz condition.

A neural network can be used to fit $f(\cdot)$. Therefore, the WGAN model turns the objective function into

$$\min_G \max_D E_{x \sim p_r}[D(x)] - E_{y \sim p_r}[D(y)] \tag{7}$$

Among them, the discriminator is required to satisfy the Lipschitz condition by using either weight cropping.

3. Methods

In this section, we present the overall framework of the proposed CWGAIN-GP model, providing a thorough explanation of the intricate design of both the generator and the discriminator. Additionally, we offer a comprehensive description of the model’s operational workflow, followed by a meticulous analysis of the convergence properties inherent to the CWGAIN-GP model.

3.1. Overview of the CWGAIN-GP model

In the collection of weather and air quality data from weather stations, certain variables may contain continuous missing values due to factors such as weather station equipment damage. During the training of the GAIN model, a data chunking method is employed to ensure a suitable learning rate, and all variables are learned sequentially based on time period divisions. When the GAIN model learns continuous missing segments, there may exist variables that are entirely missing, and the missing positions’ data are derived from other non-missing variables and the model’s iteration at that point. However,

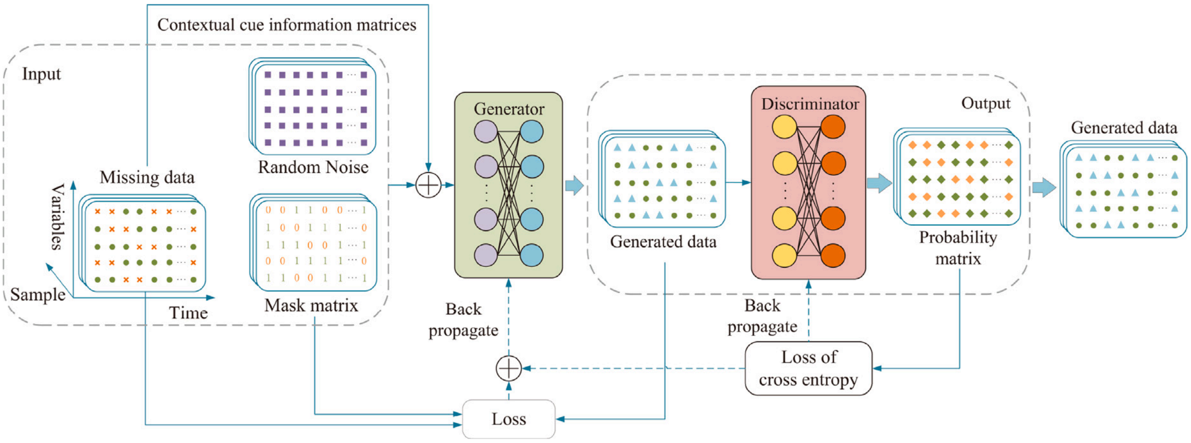


Fig. 2. Schematic diagram of the CWGAIN-GP model.

this data feature learning approach in GAIN solely relies on inter-variable correlations and lacks temporal correlation. As a consequence, the generated data might not precisely match the true values, leading to a significant discrepancy between the generated data distribution and the true distribution. This mismatch hinders the practical application of the model. Therefore, this study leverages the periodicity and strong time correlation present in time series to introduce contextual data of continuous missing values as cue information into the generator. This way aims to enhance the accuracy of imputing continuous missing values.

By incorporating the cue information matrix into the generator, the generator's multiple probability distributions exhibit less divergence, reducing the need for restrictive mechanisms like the hint mechanism in the GAIN model. Consequently, computational efforts are reduced. The CWGAIN-GP model also simplifies the neural networks of both the generator and discriminator, with both comprised of only two fully connected layers, thereby enhancing the execution efficiency of the model. Additionally, during the imputation task, if the generated values distribution and the true values distribution have no overlapping region, the discriminator becomes constant when measuring their distance using JS divergence. This leads to the problem of gradient disappearance, where error feedback fails to drive model convergence or does so very slowly. To overcome this challenge, this study adopts the Wasserstein distance, to effectively address the gradient vanishing problem by measuring the distance between generated values and true values.

Finally, the original WGAN utilizes weight clipping to enforce parameter constraints in the discriminator, ensuring Lipschitz condition. However, this method of forcibly constraining model parameters can introduce instability during training. To mitigate this issue, this study introduces an additional regularization term, namely gradient penalty, into the discriminator, replacing the weight clipping employed in the original WGAN. This modification ensures that the discriminator satisfies Lipschitz condition while avoiding the potential instability caused by weight clipping.

In summary, the structure of the CWGAIN-GP model is depicted in Fig. 2, providing a schematic representation of its architecture.

3.2. Generator structure of CWGAIN-GP model

Considering the significant loss of data information in the case of continuous missing data, a simplistic approach of pre-populating random noise into the missing positions and directly using it as the generator's input would result in a substantial deviation of the data distribution at those positions from the true value distribution. Consequently, this approach would lead to inefficiencies in training the generator. To address this issue and expedite the generator's convergence towards the true values distribution, this study employs the contextual data information cue method. Specifically, the missing data in the time series are retrieved from both preceding and subsequent time points, serving as cue information for the generator to generate data. This cue mechanism allows the generator to incorporate relevant contextual information, thereby facilitating a faster convergence towards the true values distribution.

The input to generator uses matrices X, M, N and T. The output is Y, where X is d-dimensional missing data, M is a mask matrix for X that takes the value $\{0, 1\}$, N is a d-dimensional random noise matrix where element n_{ij} takes the value range $[-0.01, +0.01]$, and T is a matrix of contextual cue messages for the generator.

Definition The matrix M and the matrix T are showed as

$$m_{ij} = \begin{cases} 0 & x_{ij} \text{ is NAN} \\ 1 & \text{otherwise} \end{cases} \tag{8}$$

$$t_{ij} = \begin{cases} x_{ij} & m_{ij} = 1 \\ x_{i\pm c,j} & m_{ij} = 0 \end{cases} \tag{9}$$

where x_{ij} is the original data i row j column element, m_{ij} is the corresponding position element of the mask matrix, and c is the length of continuous missing values at this position.

The missing data, random noise and cue information matrix are fused as

$$\bar{X} = M \odot X + (1 - M) \odot N + (1 - M) \odot T \tag{10}$$

where $\odot$ represents the multiplication of the elements in the corresponding positions of the matrix. The purpose of adding the random noise matrix N is to prevent the subsequent generator from generating data with excessive errors due to the elements with value 0 still existing in the matrix. Y is the generated result data of the generator satisfying

$$Y = G(M, \bar{X}) \tag{11}$$

3.3. Discriminator structure of CWGAIN-GP model

In the original GAN model, the discriminator's role is to determine the facticity of the data generated by the generator, providing a binary output. However, in the CWGAIN-GP model, the discriminator takes a different approach. Instead of assessing the facticity of the entire matrix as a whole, it probabilistically estimates the facticity of each element in the matrix. The discriminator discriminates whether each element is true or generated, resulting in the generation of a probability matrix. Each element in this probability matrix corresponds to an element in the same position of the input matrix. The discriminant results are then fed back to both the generator and discriminator for updating their model parameters.

The inclusion of the GP term in the CWGAIN-GP model helps address the challenges associated with weight clipping during the training process. It prevents significant deviations in model parameters and prevents the disappearance of the gradient necessary for parameter updates. This ensures that the model is consistently and effectively trained. The loss function of the discriminator, incorporating the GP term.

$$\begin{aligned} Loss_D = L_D + \nabla L_{GP} &= E_{Y\sim P_g}[D(Y)] - E_{X\sim P_r}[D(X)] \\ &\quad + \lambda E_{\tilde{X}\sim\chi}\left[\|\nabla_{\tilde{X}} D(\tilde{X})\|_2 - 1\right]^2 \end{aligned} \tag{12a}$$

$$\tilde{X} = \varepsilon M \odot X + (1-\varepsilon)(1-M)\odot Y \tag{12b}$$

where Y is the generated data sample, X is the true data sample, $\tilde{X}$ is the synthetic data using true and generated data for the convenience of solving the differential, ε is a random number from 0 to 1, P_g and P_r are the distributions of the generated and true data samples, respectively, λ is the weight of the GP term (hyper-parameter), ∇ is the differentiation factor of $G(\cdot)$, and χ is the probability distribution of $\tilde{X}$.

3.4. Algorithm for imputation of continuous missing time series by CWGAIN-GP model

During training, the CWGAIN-GP model is run according to the following steps.

(1) The extraction of training data. The samples X, M, T and N of each processing are extracted in x, m, t and n. The operation of the matrix is performed to generate $\bar{x}$ (Eq. (10)), and $\bar{x}$ and x are input to the generator to generate y (Eq. (11)). The loss function of the generator is showed as

$$Loss_G = L_G + \alpha L_{MSE} = \nabla_G[-(1-M)\odot D(Y) + \alpha M \odot (X-Y)^2] \tag{13}$$

(2) The generated result y is input to the discriminator D for discriminating. The discriminative process is showed as

$$D^*(y,m)_i = \frac{p\left(y, m_i = 1\right)}{p\left(y, m_i = 1\right) + p\left(y, m_i = 0\right)} \tag{14}$$

where $p(y,m)$ is the joint probability distribution of matrix (Y, M).

(3) The discriminant results of the discriminator are used to calculate the loss function values to optimize the discriminator and generator parameters. The Adam optimizer is chosen in this paper.

(4) The objective function of the CWGAIN-GP model can be expressed as

$$\begin{aligned} \min_G \max_D V(D,G) = E_{Y\sim P_g, \tilde{X}\sim\chi, M}[M^T D(Y) + (1-M)^T(1-D(Y)) \\ + \lambda[\|\nabla_{\tilde{X}} D(\tilde{X})\|_2 - 1]^2] \end{aligned} \tag{15}$$

where Y is the generator generated data, M is the mask matrix, $\tilde{X}$ is shown in Eq. (12b), P_g and χ are the probability distributions of the Y and $\tilde{X}$, respectively, and λ is the hyper-parameter.

(5) Return the result. After the generator generates the data and outputs Y, the recalculation of Y with the mask matrix is performed, and only the generated data is incorporated into the vacant positions of the original matrix to prevent the true data in the original data X from being destroyed. The calculation formula is showed as

$$\hat{X} = M \odot X + (1-M)\odot Y \tag{16}$$

The Algorithm 1 describes the pseudo-code of missing data imputation based on CWGAIN-GP.

Algorithm 1 Continuous Missing Data Imputation Based on CWGAIN-GP.

```
Require:
    $X$: the missing data;
    $M$: the mask matrix;
    $num\_inter$: the number of iterations;
    $extra\_d$: additional times to train the discriminator D;
    $bats$: mini-batch size;
Ensure:
    $\alpha$: hyper-parameter of generator G;
    $\lambda$: hyper-parameter of discriminator D;
For $i = 1 \rightarrow num\_iter$ do
    For $et = 1 \rightarrow extra\_d$ do
        Divide samples: $\{x(j)\}_{j=1}^{bats}$, $\{m(j)\}_{j=1}^{bats}$, $\{n(j)\}_{j=1}^{bats}$, $\{\varepsilon(j)\}_{j=1}^{bats}$ and $\{t(j)\}_{j=1}^{bats}$;
        For $j = 1 \rightarrow bats$ do
            $\bar{x}(j) \leftarrow m(j)\odot x(j) + (1-m(j))\odot n(j) + (1-m(j))\odot t(j)$;
            $y(j) \leftarrow G(m(j), \bar{x}(j))$;
            $\tilde{x}(j) \leftarrow \varepsilon(j)m(j)\odot x(j) + (1-\varepsilon(j))(1-m(j))\odot y(j)$;
        End For
        Update the discriminator D by Adam optimizer
        $\frac{1}{bats}\nabla_D \sum_{j=1}^{bats}[L_D(D(x(j)), D(y(j)), m(j)) + \lambda L_{GP}(D(\tilde{x}(j)))]$;
    End For
    Update the generator G by Adam optimizer
    $\frac{1}{bats}\nabla_D \sum_{j=1}^{bats}[-L_G(G(y(j), m(j))) + \alpha L_{MSE}(x(j), y(j), m(j))]$;
End For
return $\hat{X} \leftarrow M \odot X + (1-M)\odot \bar{Y}$
```

3.5. Convergence analysis of CWGAIN-GP

The significance of the loss function in the CWGAIN-GP model lies in its crucial role in maintaining the training stability of continuous missing time series. Achieving consistent convergence of the objective functions for both the discriminator and generator is essential to ensure that the parameters of the generator and discriminator are updated in the intended direction, effectively preventing training failures. A proof is provided below to establish the global convergence of the CWGAIN-GP objective function.

Let $\mathbb{P}_r$ and $\mathbb{P}_g$ fractions be the sample distribution and generated data distribution in the compact Metric space $\mathcal{X}$, f^* be the optimal solution of function $\max_{\|f\|_{L\le1}}\mathbb{E}_{y\sim_r}[f(y)] - \mathbb{E}_{y\sim_g}[f(y)]$ satisfying the 1-Lipschitz condition, π be the optimal coupling of $\mathbb{P}_r$ and $\mathbb{P}_g$, that is, the minimum of function $W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\pi\in\Pi(\mathbb{P}_r,\mathbb{P}_g)}\mathbb{E}_{(x,y)\sim\pi}[\|x-y\|]$, where $\Pi(\mathbb{P}_r, \mathbb{P}_g)$ is the set of joint distribution of $\pi(x,y)$, which can also be expressed as

$$\mathbb{P}_{(x,y)\sim\pi}[f^*(y) - f^*(x) = \|y-x\|] = 1 \tag{17}$$

In the vast majority of cases, the generated data are not equal to the sample data, i.e., $x \neq y$, let $\psi(t) = f^*(x_t) - f^*(x)$, and set $t, t' \in [0,1]$, which yields

$$|\psi(t) - \psi(t')| = \|f^*(x_t) - f^*(x_{t'})\| \le \|x_t - x_{t'}\| = |t-t'|\|x-y\| \tag{18}$$

So ψ is $\|x-y\|$-Lipschitz condition and the opposite leads to

$$\begin{aligned} \psi(1) - \psi(0) &= \psi(1) - \psi(t) + \psi(t) - \psi(0) \\ &\le (1-t)\|x-y\| + \psi(t) - \psi(0) \\ &\le (1-t)\|x-y\| + t\|x-y\| \\ &= \|x-y\| \end{aligned} \tag{19}$$

Since $|\psi(1) - \psi(0)| = |f^*(y) - f^*(x)| = \|y-x\|$, therefore the above Eq. (19) takes only the equal sign. In particular, $|\psi(t) - \psi(0)| = t\|x-y\|$, $\psi(0) = f^*(x) - f^*(x) = 0$, and so $\psi(t) = t\|x-y\|$. From the aforementioned information, let us derive $f^*(x_t) - f^*(x) = \psi(t) = t\|x-y\|$

from $\upsilon = \frac{y-x_t}{\|y-x_t\|} = \frac{y-((1-t)x-ty)}{\|y-((1-t)x-ty)\|} = \frac{(1-t)(y-x)}{\|(1-t)(y-x)\|} = \frac{y-x}{\|y-x\|}$, denoted as $f^*(x_t) = f^*(x)+t\|x-y\|$. By taking the partial derivative of υ, we obtain

$$\begin{aligned}\frac{\partial}{\partial \upsilon} f^*(x_t) &= \lim_{h\to 0} \frac{f^*(x_t + h\upsilon) - f^*(x_t)}{h} \\ &= \lim_{h\to 0} \frac{f^*(x + t(y-x) + \frac{h}{\|y-x\|}(y-x)) - f^*(x_t)}{h} \\ &= \lim_{h\to 0} \frac{f^*(x_{t+\frac{h}{\|y-x\|}}) - f^*(x_t)}{h} \\ &= \lim_{h\to 0} \frac{f^*(x) + (t + \frac{h}{\|y-x\|})\|x-y\| - (f^*(x_t) + t\|x-y\|)}{h} \\ &= \lim_{h\to 0} \frac{h}{h} \\ &= 1\end{aligned} \tag{20}$$

Assuming that f^* is differentiable on x_t, we can deduce $\|\nabla f^*(x_t)\| < 1$. Additionally, it is known that υ is a unit vector. By applying the Pythagorean theorem, we obtain

$$\begin{aligned}1 &\le \|\nabla f^*(x_t)\|^2 \\ &= \langle \upsilon, \nabla f^*(x_t)\rangle^2 + \|\nabla f^*(x_t) - \langle \upsilon, \nabla f^*(x_t)\rangle\,\|^2 \\ &= |\frac{\partial}{\partial \upsilon} f^*(x_t)|^2 + \|\nabla f^*(x_t) - \upsilon\frac{\partial}{\partial \upsilon} f^*(x_t)\|^2 \\ &= 1 + \|\nabla f^*(x_t) - \upsilon\|^2 \le 1\end{aligned} \tag{21}$$

So Eq. (21) takes only the equal sign and yields $\|\nabla f^*(x_t) - \upsilon\|^2 = 0$, which is $\nabla f^*(x_t) = \upsilon = \frac{y-x_t}{\|y-x_t\|}$.

In summary, we can conclude that if (x, y) satisfies $f^*(y) - f^*(x) = \|y - x\|$, it implies $\nabla f^*(x_t) = \upsilon = \frac{y-x_t}{\|y-x_t\|}$. Furthermore, under the condition of π, where the probability is 1, we get

$$\mathbb{P}_{(x,y)\sim\pi}\left[\nabla f^*(x_t) = \frac{y-x_t}{\|y-x_t\|}\right] = 1 \tag{22}$$

Therefore, the proposed CWGAIN-GP objective function possesses an optimal solution irrespective of the coupling between the generated and sample data, thereby guaranteeing the convergence of the model training process.

4. Experiments and results

In this section, the proposed CWGAIN-GP is evaluated on two real-world public datasets. The first part is about the datasets preparation and algorithm implementation details including test environment, model settings and hyper-parameters. Then we evaluate the CWGAIN-GP imputation method on the datasets comparing with baseline methods to highlight the state-of-the-art (SOTA) performance of our proposed model. The third part presents the ablation study of the proposed CWGAIN-GP model. Finally, we study the effect of different lengths of continuous missing values on the model's imputation ability.

4.1. Datasets description

(1) Beijing Air Quality and Meteorological Dataset[1]: This dataset is an open, real-world dataset from the UCI [37]. The dataset consists of six air quality indices: PM2.5, PM10, SO2, NO2, CO, and O3, along with four meteorological measurements: temperature (T), barometric pressure (P), dew point temperature (DEWP), and wind speed (WS). Hourly data for all variables were collected from March 2013 to February 2017, totaling 35064 datasets (for the convenience of narration, we will call it beijing dataset for short).

(2) KDD CUP 2018 Dataset[2]: This dataset is a public air quality dataset from the KDD CUP Challenge 2018 . For this paper, we selected the weather data for London from the year 2020, recorded at a frequency of once every 10 min. It includes 15 variables such as barometric pressure, Celsius temperature, specific humidity, relative humidity, dew-point temperature, wind speed, and more. The dataset comprises a total of 52696 data sets for the entire year (we will call it KDD dataset later).

The datasets mentioned above undergo data feature visualization to illustrate the motivation behind the feature correlation mechanism. Heat maps are created to display the average Pearson correlation coefficient between any two feature vectors within each dataset. For the Beijing dataset, Fig. 3(a) illustrates the average Pearson correlation coefficients among its variables. This heat map demonstrates strong correlations among the variables, with positive correlations observed between PM2.5, PM10, SO2, NO2, and CO, and predominantly negative correlations between O3, T, P, DEWP, and WS. Similarly, for the KDD dataset, Fig. 3(b) presents the heat map of the average Pearson correlation coefficients among the 16 variables. This heat map reveals noticeable positive and negative correlations among some of the variables. Both datasets exhibit complex inter-variable correlations, necessitating a model with strong feature-capturing capabilities. Understanding these patterns of interactions between different features is crucial for achieving accurate missing values imputation.

4.2. Implementation details

To generate the missing experimental data, the datasets is initially divided into several regions. The number of missing regions is determined based on the specified missing rate. Subsequently, these regions are randomly assigned as missing regions according to the defined missing rate, resulting in the creation of continuous missing time series data.

During the training of the CWGAIN-GP model, the Adam optimizer is employed in this study. The learning rates for both the generator and discriminator are grid searched within the range of 0.01, 0.001, 0.0001, 0.00001. The best performance of the CWGAIN-GP model is achieved when imputing the Beijing dataset using a learning rate of 0.001. The minimum length of the training batch data is set to 128. Furthermore, hyper-parameters for the discriminator, denoted as λ, are explored within the range of 5, 10, 12, 20, while hyper-parameters for the generator, denoted as α, are explored within the range of 50, 80, 100, 120. The best results for the Beijing and KDD dataset are obtained when $\lambda = 12$, $\alpha = 100$ and $\lambda = 10$, $\alpha = 120$. The training process is conducted for a total of 10,000 iterations.

The root mean squared error (RMSE), symmetric mean absolute percentage error (SMAPE) and mean absolute scaled error (MASE) are chosen for the imputation error measurement. When performing the RMSE error calculation, it is necessary to ensure that the weights of each variable are equal, so each variable data should be normalized before comparing the imputed data $\hat{X}$ with the true data X.

For the simulation experiments, the following test environment has been selected: an Intel(R) Core(TM) i5-7400 CPU running at 3.00 GHz, 24 GB of RAM, the Windows 10 platform, Python 3.7, and the TensorFlow environment.

4.3. Baseline methods for comparison

To reflect the better imputation accuracy of CWGAIN-GP model on the continuous missing of time series, several commonly used time series imputation methods, including statistical-based methods, machine learning methods and neural network methods, are selected for comparison in this section.

[1] https://archive.ics.uci.edu/dataset/501/beijing+multi+site+air+quality+data

[2] https://www.biendata.xyz/competition/kdd_2018/

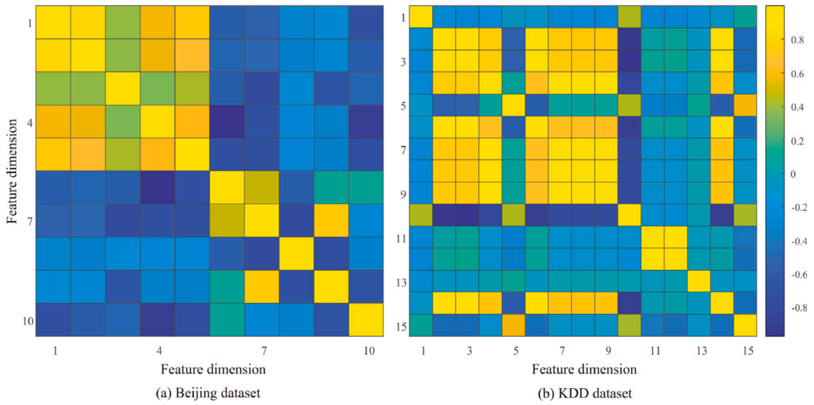


Fig. 3. Visualization of correlation matrix of data features in Beijing region.

(1) Traditional statistical methods: Mean method imputes missing data by using the global mean of historical observations. MICE predicts missing data through a series of iterations. Softimpute is an algorithm that estimates missing data through a soft thresholding process of singular value decomposition, primarily designed for missing large-scale matrix data [38]. K- nearest neighbor(KNN) estimates missing values by leveraging the weighted results of neighboring samples with high similarity [39].

(2) Machine learning methods: Random forest (RF) utilizes the features of each missing value as labels for estimation and imputation. It iteratively models each feature with missing values as a function of other features to complete the estimation of missing values.

(3) Neural network methods: BRITS is a recurrent neural network-based imputation method [16]. It incorporates BRNN structures and combines the decay mechanism with recurrent dynamical systems for missing data estimation. SAITS is based on the self-attention mechanism (SAITS) [17]. It is trained using a joint optimization method and learns data features from the weighted combination of two diagonally masked self-attention blocks for missing value imputation. GAIN models are based on the GAN architecture [25].

These methods are selected as baselines for comparison with the CWGAIN-GP model to assess its imputation performance.

4.4. Experimental results

4.4.1. Imputation results

In the real-world, machine examines and repairs may span several hours or a day or two, and the missing rates in time series data can vary in magnitude. To account for these real-world variations, this study employs a simulation approach, randomly masking some measurements within the time series as missing values. However, it is important to note that when the missing rate becomes excessively high, the data loses its practical significance for imputation purposes. As a result, the study focuses on evaluating imputation performance under more realistic conditions, specifically at missing rates of 10%, 20%, 30%, 40%, and 50%. Furthermore, the length of continuous missing values is consistently set to 100.

Table 1 presents the imputation results based on statistical, machine learning, neural network, and the proposed CWGAIN-GP models for different missing rates. From the table, it can be observed that statistical-based imputation methods, such as mean imputation and KNN, generally perform poorly, with KNN being the least effective among all methods. On the other hand, MICE imputation shows relatively good performance, possibly due to its ability to capture the correlation between different feature dimensions in the dataset. Overall, the errors of all three neural network-based methods are significantly lower than those of the statistical-based methods (Mean, Softimpute, and KNN), indicating the superiority of neural network approaches in imputation.

Comparing the complete CWGAIN-GP model with the statistical models, machine learning models, and neural network models, it can be seen that the proposed model outperforms the statistical-based and machine learning imputation methods in terms of RMSE, SMAPE and MASE metrics, and slightly outperforms the neural network imputation model. This demonstrates the advantages of generative adversarial networks in missing data imputation. Furthermore, the imputation error of the CWGAIN-GP model increases with the increase of the missing rate, which can be attributed to the model capturing less valid information as the missing rate becomes higher, leading to a degradation in performance. In conclusion, regardless of variations in the missing rate, the proposed CWGAIN-GP model consistently outperforms other models, which showing its effectiveness.

Fig. 4 presents the error distribution of the CWGAIN-GP model for imputed values and true values of Beijing datastes. The data are normalized and differences are calculated, and 1000 missing locations are randomly selected. It can be observed that the errors on different missing rates exhibit a normal distribution, indicating random errors and the realism of the experiment.

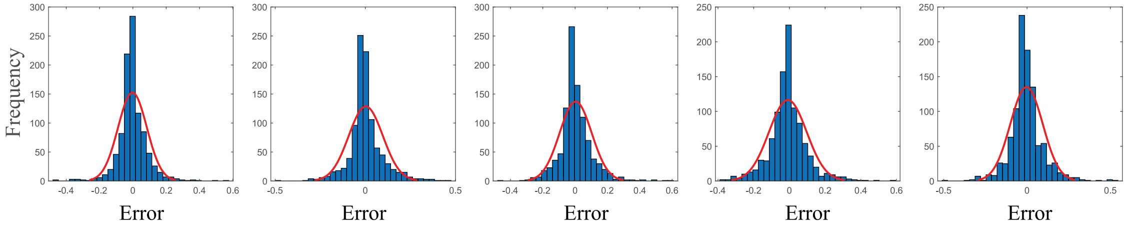
In order to provide additional substantiation of the proposed model's capacity for generalization, a replication of pertinent experiments was undertaken employing the KDD dataset. The results of these experiments in Table 2, present the outcomes of three distinct evaluation metrics applied to various models that engage in imputation with the KDD dataset. Evidently, the superior performance of our model is manifest as it consistently attains the lowest error rate.

To provide a visual representation of these distinctions, Fig. 5 visually illustrates the values presented in Table 1 and Table 2. Within this depiction, the horizontal axis corresponds to the different models employed, while the vertical axis represents the error metrics. Furthermore, distinct colors are employed to signify varying degrees of missing data. It can be seen that the errors of most of the models have an overall increasing trend with increasing missing rate, which is in line with the empirical knowledge. Of noteworthy significance is the fact that our model consistently achieves the most favorable performance, yielding the minimum error across the majority of the spectrum of missing data rates.

To facilitate a clearer and more intuitive comparison of the imputation capabilities of each model, statistical tests were conducted on the error data presented in Tables 1 and 2, and the results are visually represented in Fig. 6. In this context, the center of the distinctively colored line segments convey the model's relative imputation ranking, and the closer to the left indicates the higher ranking. The length of the line segments represents the critical difference, and the less overlap between different line segments indicates the more significant difference in model imputation ability. In this comparative analysis, we first performed the Friedman test at a confidence level of 0.05 and obtained a critical difference of 2.80. The test results showed that

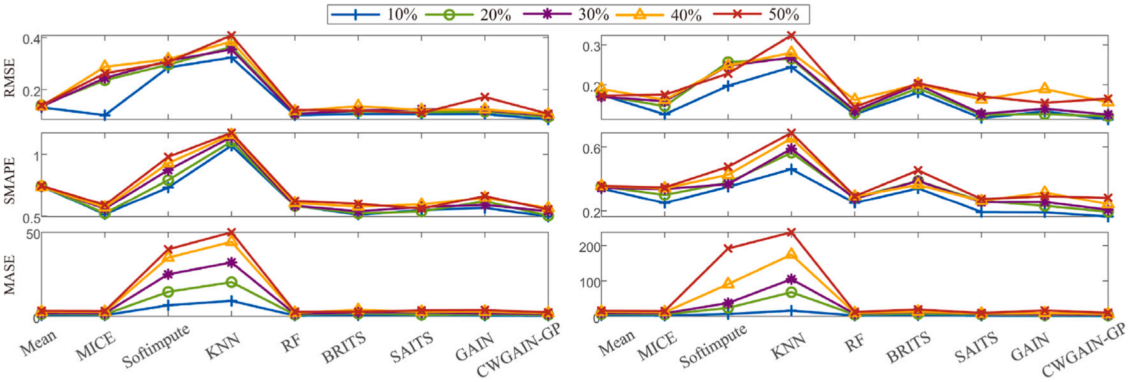
Table 1
Comparison of imputation results on different missing rates.

Model		Error	Missing rates				
			10%	20%	30%	40%	50%
Statistical models	Mean	RMSE	0.1309	0.1390	0.1343	0.1352	0.1389
		SMAPE	0.7422	0.7450	0.7394	0.7381	0.7465
		MASE	0.6152	1.3357	1.9238	2.6043	3.2405
	MICE	RMSE	0.1009	0.2355	0.2440	0.2871	0.2617
		SMAPE	0.5155	0.5222	0.5597	0.5708	0.5912
		MASE	0.6158	1.1072	1.7603	2.5886	3.1487
	Softimpute	RMSE	0.2847	0.2954	0.3105	0.3164	0.3051
		SMAPE	0.7323	0.7925	0.8727	0.9279	0.9788
		MASE	6.5439	14.543	24.974	34.875	39.819
Machine learning models	KNN	RMSE	0.3229	0.3634	0.3540	0.3845	0.4081
		SMAPE	1.0692	1.1014	1.1379	1.1555	1.1711
		MASE	9.1579	20.305	32.031	44.302	49.968
	RF	RMSE	0.1011	0.1072	0.1058	0.1168	0.1210
		SMAPE	0.5847	0.5818	0.5833	0.6056	0.6231
		MASE	0.4448	0.9559	1.3899	2.1183	2.6879
Neural Networks models	BRITS	RMSE	0.1052	0.1159	0.1189	0.1359	0.1188
		SMAPE	0.5107	0.5217	0.5393	0.5772	0.5994
		MASE	0.6378	1.6401	2.3578	3.7739	2.6523
	SAITS	RMSE	0.1049	0.1123	0.1241	0.1207	0.1098
		SMAPE	0.5483	0.5379	0.5729	0.5944	0.5608
		MASE	0.6790	0.9142	2.3615	2.6435	3.5241
	GAIN	RMSE	0.1051	0.1145	0.1202	0.1231	0.1710
		SMAPE	0.5664	0.6197	0.5904	0.6491	0.6632
		MASE	0.4733	1.0544	1.5963	2.7270	3.7012
Our Model	CWGAIN-GP	RMSE	**0.0845**	**0.0946**	**0.0997**	**0.1037**	**0.1062**
		SMAPE	**0.4965**	**0.5033**	**0.5387**	**0.5646**	**0.5531**
		MASE	**0.3977**	**0.8816**	**1.3369**	**1.9678**	**2.3287**



(a)10% (b)20% (c)30% (d)40% (e)50%

Fig. 4. Imputation error distribution of CWGAIN-GP model.



(a) Beijing datasets (b) KDD datasets

Fig. 5. Imputation results (in RMSE, SMAPE and MASE) of Beijing and KDD dataset.

there was a difference in performance between the models, and then we employed the Nemenyi test as a post-hoc to further distinguish between the algorithms. As observed in Fig. 6, our model (indicated by the yellow line) consistently aligns closest to the left on both datasets. This positioning underscores its superior performance in terms of imputation ability on two datasets, although the margin of advantage relative to some other models may not be overwhelmingly pronounced.

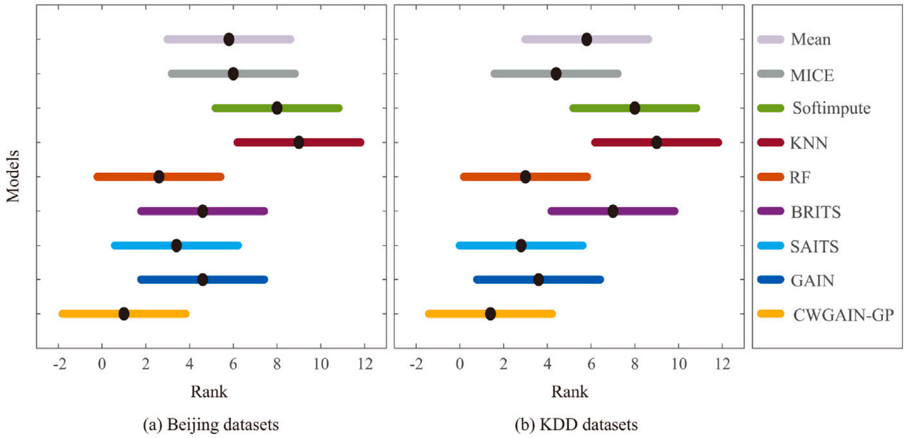
4.4.2. *Ablation study*

To assess the impact of the proposed model's innovations and demonstrate the magnitude of improvement in imputation performance, three ablation study are conducted in this paper. The details of these ablation study are as follows:

(1) Removing the contextual cue matrix (WGAIN-GP): In this experiment, only the contextual cue information matrix of the input

Table 2
Comparison of imputation results on different missing rates.

Model		Error	Missing rates				
			10%	20%	30%	40%	50%
Statistical models	Mean	RMSE	0.1747	0.1732	0.1696	0.1894	0.1727
		SMAPE	0.3409	0.3542	0.3445	0.3564	0.3578
		MASE	3.1093	6.5463	9.6348	12.556	15.792
	MICE	RMSE	0.1258	0.1464	0.1587	0.1621	0.1751
		SMAPE	0.2494	0.3002	0.3375	0.3420	0.3473
		MASE	2.0234	5.1404	8.3884	11.673	14.937
	Softimpute	RMSE	0.1973	0.2567	0.2475	0.2479	0.2279
		SMAPE	0.3514	0.3783	0.3685	0.4267	0.4763
		MASE	6.1421	23.671	37.483	90.945	191.79
Machine learning models	KNN	RMSE	0.2441	0.2651	0.2681	0.2801	0.3236
		SMAPE	0.4611	0.5634	0.5879	0.6527	0.6868
		MASE	15.996	68.096	105.03	174.67	237.90
	RF	RMSE	0.1261	0.1294	0.1348	0.1619	0.1641
		SMAPE	0.2500	0.2843	0.2771	0.2912	0.2928
		MASE	2.0270	4.5158	7.0529	9.4491	12.350
Neural Networks models	BRITS	RMSE	0.1804	0.1914	0.2004	0.2013	0.2037
		SMAPE	0.3419	0.3884	0.3875	0.3639	0.4540
		MASE	3.2460	7.0164	12.362	13.007	19.305
	SAITS	RMSE	0.1164	0.1244	0.1271	0.1631	0.1707
		SMAPE	0.1926	0.2622	0.2547	0.2625	0.2846
		MASE	1.9682	4.9366	6.2172	7.8093	9.9086
	GAIN	RMSE	0.1336	0.1263	0.1397	0.1892	**0.1547**
		SMAPE	0.1914	0.2315	0.2554	0.3156	0.2919
		MASE	1.5773	3.4283	5.1926	9.0604	16.524
Our Model	CWGAIN-GP	RMSE	**0.1130**	**0.1213**	**0.1252**	**0.1561**	0.1652
		SMAPE	**0.1643**	**0.1944**	**0.2066**	**0.2420**	**0.2825**
		MASE	**0.9899**	**2.4756**	**4.5062**	**6.3817**	**9.8203**



(a) Beijing datasets

(b) KDD datasets

Fig. 6. Friedman test for models imputation performance.

Table 3
Imputation results of ablation study on Beijing dataset.

Model	Error	Missing rates				
		10%	20%	30%	40%	50%
WGAIN-GP	RMSE	0.0970	0.1284	0.1278	0.1312	0.1751
	SMAPE	0.5953	0.6795	0.6847	0.7113	0.8031
	MASE	0.5125	1.3239	1.9914	2.8521	4.1651
CGAIN-GP	RMSE	0.0937	0.1227	0.1457	0.1153	0.1159
	SMAPE	0.5143	0.5745	0.6337	0.5913	0.5671
	MASE	0.4122	0.9647	1.5216	2.0999	2.3606
CWGAIN	RMSE	0.0875	0.0959	**0.0942**	0.1283	0.1193
	SMAPE	0.5021	0.5059	**0.5305**	0.8204	0.5772
	MASE	0.4061	0.8969	1.3582	2.0292	2.4837
CWGAIN-GP	RMSE	**0.0845**	**0.0946**	0.0997	**0.1037**	**0.1062**
	SMAPE	**0.4965**	**0.5033**	0.5387	**0.5646**	**0.5531**
	MASE	**0.3977**	**0.8816**	**1.3369**	**1.9678**	**2.3287**

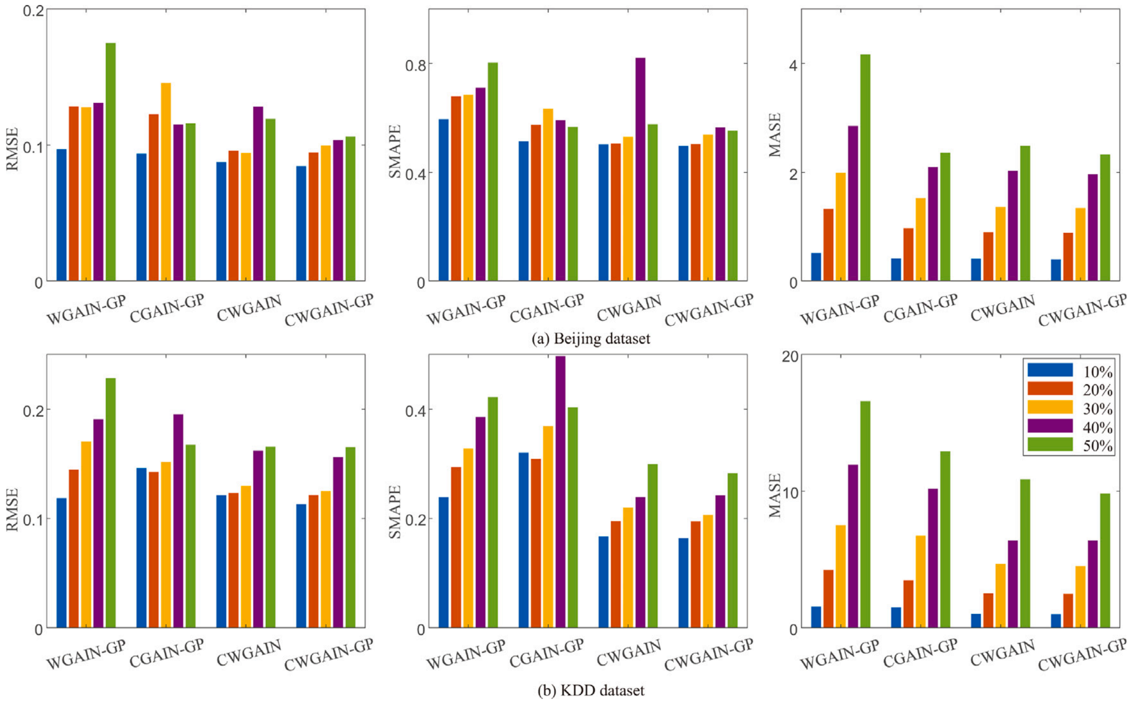
generator in the CWGAIN-GP model is removed. The loss function and the structure between each module of the model remain unchanged.

(2) No Wasserstein distance is used (CGAIN-GP): In this experiment, the discriminator structure of the original GAN model is utilized, and the constraint on the contextual cue matrix is maintained. While the GP is retained in the discriminator, the Lipschitz condition on the loss function value is eliminated.

(3) Remove the GP (CWGAIN): In this experiment, only the GP term in the discriminator's loss function is removed, while the rest of the CWGAIN-GP model remains unchanged.

Tables 3 and 4 demonstrate the imputation errors with the WGAIN-GP, CGAIN-GP, CWGAIN, and the model proposed in this study, on the Beijing and KDD dataset.

In Table 3, it is readily apparent that the imputation error of WGAIN-GP ranks as the most substantial. Moreover, it is observed that the error of CWGAIN slightly surpasses that of CWGAIN-GP. This observation means that the proposed model plays a dominant role in performing the task of imputing continuous missing values with the contextual cue matrix constraints strategy, and the loss function using the Wasserstein distance and GP terms provides a more limited enhancement to the model.



(a) Beijing dataset

(b) KDD dataset

Fig. 7. Imputation results (in RMSE, SMAPE and MASE) of ablation study on different missing rates.

Table 4
Imputation results of ablation study on KDD dataset.

Model	Error	Missing rates				
		10%	20%	30%	40%	50%
	RMSE	0.1185	0.1445	0.1705	0.1908	0.2283
WGAIN-GP	SMAPE	0.2385	0.2940	0.3276	0.3854	0.4218
	MASE	1.5521	4.2304	7.5187	11.943	16.578
	RMSE	0.1460	0.1423	0.1516	0.1953	0.1673
CGAIN-GP	SMAPE	0.3202	0.3086	0.3687	0.4972	0.4034
	MASE	1.5038	3.4746	6.7409	10.188	12.904
	RMSE	0.1212	0.1232	0.1298	0.1620	0.1657
CWGAIN	SMAPE	0.1673	0.1949	0.2197	**0.2386**	0.2993
	MASE	1.0126	2.5138	4.6757	6.3875	10.865
	RMSE	**0.1130**	**0.1213**	**0.1252**	**0.1561**	**0.1652**
CWGAIN-GP	SMAPE	**0.1643**	**0.1944**	**0.2066**	0.2420	**0.2825**
	MASE	**0.9899**	**2.4756**	**4.5062**	**6.3817**	**9.8203**

Table 5
Imputation results of Beijing dataset on different lengths of missing values.

Model	Error	Lengths of continuous missing values				
		1	10	50	100	200
	RMSE	0.1364	0.1374	0.1375	0.1343	0.1365
Mean	SMAPE	0.7390	0.7429	0.7405	0.7394	0.7592
	MASE	1.9384	1.9537	1.9800	1.9238	1.9478
	RMSE	0.4079	0.3803	0.3732	0.3540	0.3542
KNN	SMAPE	1.0874	1.0995	1.1048	1.1379	1.0734
	MASE	30.391	31.027	32.243	32.031	29.318
	RMSE	0.1026	0.1295	0.1322	0.1202	0.1251
GAIN	SMAPE	0.5728	0.6525	0.7135	0.5904	0.6723
	MASE	1.6450	6.6143	2.4482	1.5963	1.9205
	RMSE	**0.0565**	**0.1029**	**0.0967**	**0.0997**	**0.1066**
CWGAIN-GP	SMAPE	**0.3773**	**0.4947**	**0.5257**	**0.5387**	**0.5563**
	MASE	**0.9870**	**1.2972**	**1.3698**	**1.3369**	**1.3542**

In Table 4, CWGAIN emerges as the model with the most minimal error at missing rates of 10%, 20%, and 30% for the KDD dataset, whereas CGAIN-GP has the smallest error at 40% and 50% missing rates. This suggests that the GP term exerts the least influence on the model's performance when confronted with lower missing data rates, while the Wasserstein distance feature yields the least enhancement when handling higher levels of missing data. We visualize the data from Tables 3 and 4 as shown in Fig. 7. Overall, the complete CWGAIN-GP model proposed in this paper imputes the data best under various missing rates, which also indicates that the three improvements are effective in improving the imputation task.

By conducting these ablation experiments, the paper aims to evaluate the contribution of each innovation in the proposed model and quantify the improvement in imputation performance achieved by these innovations.

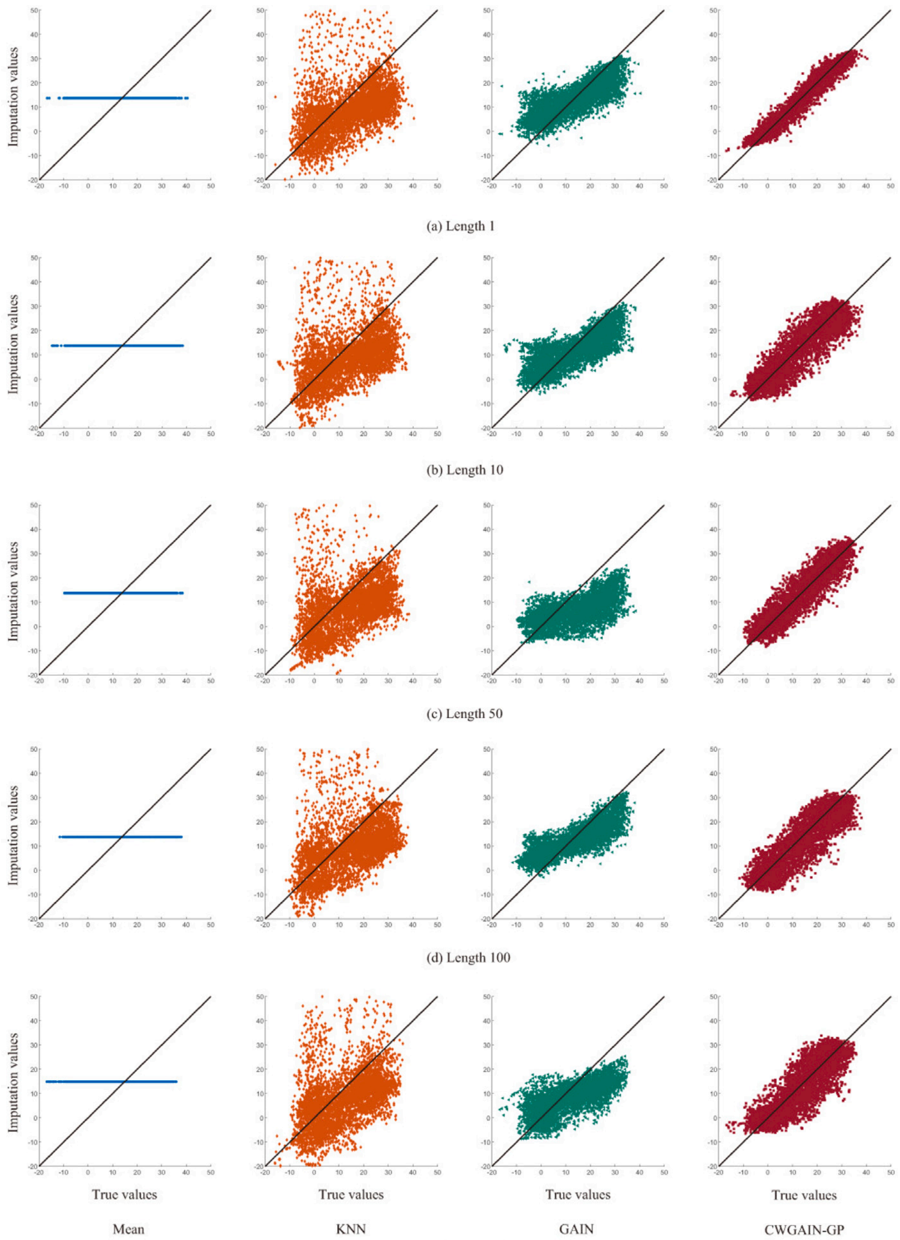
4.4.3. A study of the effect of the lengths of continuous missing values

The time series data with continuous missing patterns not only have different missing rates, but also have differences in the lengths of continuous missing. In order to make the experiment more convincing, this study introduces different lengths of continuous missing values within both Beijing and KDD dataset to further demonstrate the effectiveness of the proposed model for imputation continuous missing values in time series. In order to weigh the effects of both high and low missing rates, we have set lengths of continuous missing at 1, 10, 50, 100, 200 while maintaining a middle missing rate of 30%. Subsequently, we have conducted comparative experiments, pitting the proposed model against the representative models in Section 4.4.1. The results are shown in Tables 5 and 6.

From Table 5, it can be observed that the imputation errors of the proposed CWGAIN-GP model are generally smaller than those of the other methods. This indicates that the CWGAIN-GP model possesses effective data information extraction capabilities and exhibits superior performance in imputing missing values for both long and short lengths of continuous missing values.

4.4.4. A study of the correlation between imputed and true values

In contrast to error metrics such as RMSE, the utilization of visual plots to depict the correlation and difference between true values and imputed values provides a more lucid depiction of the characteristics of the imputed data. For example, in Section 4.4.3, although the error of the mean imputation method is slightly worse than the proposed



(a) Length 1

(b) Length 10

(c) Length 50

(d) Length 100

(e) Length 200

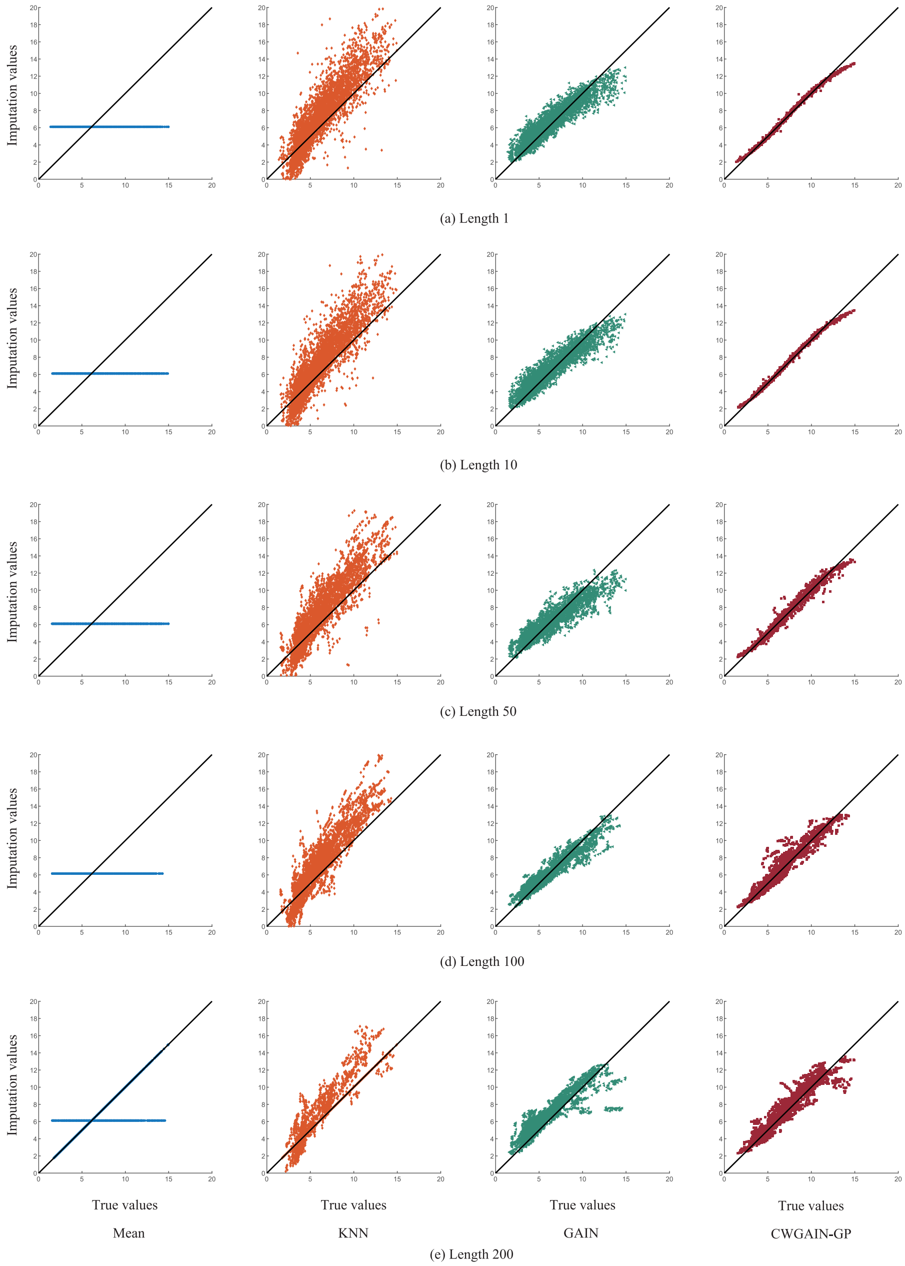
Fig. 8. Correlations between true and imputed values of model imputed Beijing dataset.

Table 6
Imputation results of KDD dataset on different lengths of missing values.

Model	Error	Lengths of continuous missing values				
		1	10	50	100	200
	RMSE	0.1719	0.1728	0.1728	0.1696	0.1714
Mean	SMAPE	0.3480	0.3492	0.3473	0.3445	0.3511
	MASE	9.5587	9.6642	9.7044	9.6348	10.044
	RMSE	0.3154	0.2656	0.2586	0.2681	0.2694
KNN	SMAPE	0.6057	0.6042	0.6069	0.5879	0.6020
	MASE	108.33	107.50	122.83	105.03	103.17
	RMSE	0.1171	0.1215	0.1211	0.1397	**0.1155**
GAIN	SMAPE	0.2250	0.2332	0.2133	0.2554	0.2499
	MASE	5.1381	5.3280	4.8271	5.1926	5.4275
	RMSE	**0.0539**	**0.0807**	**0.1076**	**0.1252**	0.1297
CWGAIN-GP	SMAPE	**0.0923**	**0.1262**	**0.1838**	**0.2066**	**0.2381**
	MASE	**1.1272**	**1.5698**	**3.2937**	**4.5062**	**5.1414**

method, the true distribution of the imputed data is a uniform distribution of constant values, which is completely different from the true values distribution. Thus, this paper embraces the insights from Section 4.4.3, visually comparing the imputed data with the true values, with the resultant graphical representation furnished in Fig. 8 and Fig. 9. The different colors, progressing from left to right, correspond to distinct models, while variations from top to bottom denote differing lengths of continuous missing. The horizontal coordinates denote the true values, whereas the vertical coordinates signify the imputed values. Consequently, data dots closer to the positive linear regression curve indicate superior model imputation efficacy.

It is worth noting that the imputed values rendered by the CWGAIN-GP model, particularly evident in the KDD dataset, closely adhere to the positive linear regression curve. This striking performance underscores the exceptional imputation capabilities of the CWGAIN-GP model, coupled with the smooth data value fluctuations within the KDD dataset. A compelling comparison can be made with the CWGAIN-GP model depicted in Fig. 8 (red dots), which consistently aligns more closely to



(a) Length 1

(b) Length 10

(c) Length 50

(d) Length 100

(e) Length 200

Fig. 9. Correlations between true and imputed values of model imputed KDD dataset.

the positive linear regression line in contrast to the GAIN model (green dots), across varying lengths of continuous missing.

Furthermore, the imputation results for the GAIN model in Fig. 9 reveal that GAIN's data dots tend to exhibit lateral skewness, notably at continuous missing lengths of 50, 100, and 200. The KNN method's imputed values (orange dots) manifest pronounced deviations from the true data, contributing to significantly larger error metrics compared to other models. The imputed and true values of the CWGAIN-GP model, both for long and short lengths of continuous missing, exhibit a strong and favorable correlation, reinforcing its imputation prowess.

5. Conclusion and future work

In this paper, we propose a new imputation model based on GAN to solve the continuous missing problem in time series. The proposed CWGAIN-GP model introduces a contextual cue information matrix to constrain the range of values generated in order to avoid the problem of biased distribution of generated data due to too few local features of the continuously missing data, and optimizes the mathematical model by improving the loss function of the model to improve the modeling capability of the model. To evaluate the imputation performance of the proposed model, a series of experiments involving imputation continuous missing values at missing rates ranging from 10% to 50% are conducted on publicly available datasets, including Beijing air quality and meteorological dataset and KDD CUP 2018 dataset. The study assesses the imputation capabilities of three type imputation models, demonstrated how the improvement points of the proposed model affect the estimation task. In addition, we also compare the imputation accuracy between models on different lengths of continuous missing, as well as a visual comparison of the correlation between the imputed and true values. The experimental results show that the proposed CWGAIN-GP model exhibits better imputation performance than the existing various types of imputation models.

Nonetheless, it is essential to acknowledge that our proposed CWGAIN-GP model does have certain limitations. Firstly, it currently lacks the capacity to simultaneously train on multiple datasets inputs, thereby limiting its ability training geographic information. Secondly, the generator component of the proposed model is not optimized for discrete values and can exclusively generate continuous values. In essence, our model does not support the imputation of discrete data.

CRediT authorship contribution statement

Yunsheng Wang: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Conceptualization. **Xinghan Xu:** Writing – review & editing, Supervision, Data curation. **Lei Hu:** Writing – review & editing, Visualization. **Jianchao Fan:** Writing – review & editing, Project administration. **Min Han:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62173063).

References

[1] L. Bai, L. Yao, C. Li, X. Wang, C. Wang, Adaptive graph convolutional recurrent network for traffic forecasting, in: Advances in Neural Information Processing Systems, Vol. 33, Curran Associates, Inc., 2020, pp. 17804–17815.

[2] S.J. Choudhury, N.R. Pal, Imputation of missing data with neural networks for classification, Knowl.-Based Syst. 182 (2019) 104838.

[3] J. Li, W. Ren, M. Han, Variational auto-encoders based on the shift correction for imputation of specific missing in multivariate time series, Measurement 186 (2021) 110055.

[4] Y. Zhang, P.J. Thorburn, Handling missing data in near real-time environmental monitoring: A system and a review of selected methods, Future Gener. Comput. Syst. 128 (2022) 63–72.

[5] B. Yang, Y. Kang, Y. Yuan, X. Huang, H. Li, ST-LBAGAN: Spatio-temporal learnable bidirectional attention generative adversarial networks for missing traffic data imputation, Knowl.-Based Syst. 215 (2021) 106705.

[6] X. Miao, Y. Wu, L. Chen, Y. Gao, J. Yin, An experimental survey of missing data imputation algorithms, IEEE Trans. Knowl. Data Eng. 35 (7) (2023) 6630–6650.

[7] D. Li, L. Li, X. Li, Z. Ke, Q. Hu, Smoothed LSTM-AE: A spatio-temporal deep model for multiple time-series missing imputation, Neurocomputing 411 (2020) 351–363.

[8] C. Sun, Y. Chen, C. Cheng, Imputation of missing data from offshore wind farms using spatio-temporal correlation and feature correlation, Energy 229 (2021) 120777.

[9] Z. Pan, Y. Wang, K. Wang, H. Chen, C. Yang, W. Gui, Imputation of missing values in time series using an adaptive-learned median-filled deep autoencoder, IEEE Trans. Cybern. 53 (2) (2023) 695–706.

[10] W.-C. Lin, C.-F. Tsai, J.R. Zhong, Deep learning for missing value imputation of continuous data and the effect of data discretization, Knowl.-Based Syst. 239 (2022) 108079.

[11] Q. Ma, S. Li, L. Shen, J. Wang, J. Wei, Z. Yu, G.W. Cottrell, End-to-end incomplete time-series modeling from linear memory of latent variables, IEEE Trans. Cybern. 50 (12) (2020) 4908–4920.

[12] M.D. Samad, S. Abrar, N. Diawara, Missing value estimation using clustering and deep learning within multiple imputation framework, Knowl.-Based Syst. 249 (2022) 108968.

[13] W. Zhang, P. Zhang, Y. Yu, X. Li, S.A. Biancardo, J. Zhang, Missing data repairs for traffic flow with self-attention generative adversarial imputation net, IEEE Trans. Intell. Transp. Syst. 23 (7) (2022) 7919–7930.

[14] Q. Ma, S. Li, G.W. Cottrell, Adversarial joint-learning recurrent neural network for incomplete time series classification, IEEE Trans. Pattern Anal. Mach. Intell. 44 (4) (2022) 1765–1776.

[15] X. Zhou, X. Liu, G. Lan, J. Wu, Federated conditional generative adversarial nets imputation method for air quality missing data, Knowl.-Based Syst. 228 (2021) 107261.

[16] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, Y. Li, BRITS: Bidirectional recurrent imputation for time series, in: S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 31, Curran Associates, Inc., 2018.

[17] W. Du, D. Côté, Y. Liu, SAITS: Self-attention-based imputation for time series, Expert Syst. Appl. 219 (2023) 119619.

[18] Y. Luo, X. Cai, Y. ZHANG, J. Xu, Y. xiaojie, Multivariate time series imputation with generative adversarial networks, in: S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 31, Curran Associates, Inc., 2018.

[19] Q. Ma, X. Li, M. Bai, X. Wang, B. Ning, G. Li, MIVAE: Multiple imputation based on variational auto-encoder, Eng. Appl. Artif. Intell. 123 (2023) 106270.

[20] Z. Xu, S. Wu, Q. Jiao, H.-S. Wong, TSEV-GAN: Generative adversarial networks with target-aware style encoding and verification for facial makeup transfer, Knowl.-Based Syst. 257 (2022) 109958.

[21] S.E. Awan, M. Bennamoun, F. Sohel, F. Sanfilippo, G. Dwivedi, Imputation of missing data with class imbalance using conditional generative adversarial networks, Neurocomputing 453 (2021) 164–171.

[22] S. Yang, M. Dong, Y. Wang, C. Xu, Adversarial recurrent time series imputation, IEEE Trans. Neural Netw. Learn. Syst. 34 (4) (2023) 1639–1650.

[23] G. Boquet, J.L. Vicario, A. Morell, J. Serrano, Missing data in traffic estimation: A variational autoencoder imputation method, in: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, 2019, pp. 2882–2886.

[24] Z. Guo, Y. Wan, H. Ye, A data imputation method for multivariate time series based on generative adversarial network, Neurocomputing 360 (2019) 185–197.

[25] J. Yoon, J. Jordon, M. van der Schaar, GAIN: Missing data imputation using generative adversarial nets, in: J. Dy, A. Krause (Eds.), Proceedings of the 35th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 80, PMLR, 2018, pp. 5689–5698.

[26] Y. Luo, Y. Zhang, X. Cai, X. Yuan, E2gan: End-to-end generative adversarial network for multivariate time series imputation, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, AAAI Press, 2019, pp. 3094–3100.

[27] X. Lai, X. Wu, L. Zhang, Autoencoder-based multi-task learning for imputation and classification of incomplete data, Appl. Soft Comput. 98 (2021) 106838.

[28] Q. Ni, X. Cao, MBGAN: An improved generative adversarial network with multi-head self-attention and bidirectional RNN for time series imputation, Eng. Appl. Artif. Intell. 115 (2022) 105232.

[29] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in: D. Precup, Y.W. Teh (Eds.), Proceedings of the 34th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 70, PMLR, 2017, pp. 214–223.

[30] M. Zhang, H. Wang, P. He, A. Malik, H. Liu, Exposing unseen GAN-generated image using unsupervised domain adaptation, Knowl.-Based Syst. 257 (2022) 109905.

[31] C. Wang, C. Xu, X. Yao, D. Tao, Evolutionary generative adversarial networks, IEEE Trans. Evol. Comput. 23 (6) (2019) 921–934.

[32] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, D.N. Metaxas, StackGAN++: Realistic image synthesis with stacked generative adversarial networks, IEEE Trans. Pattern Anal. Mach. Intell. 41 (8) (2019) 1947–1962.

[33] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, Commun. ACM 63 (11) (2020) 139–144.

[34] S. Liu, K. Tang, X. Yao, Generative adversarial construction of parallel portfolios, IEEE Trans. Cybern. 52 (2) (2022) 784–795.

[35] X. Na, W. Ren, M. Liu, M. Han, Hierarchical echo state network with sparse learning: A method for multidimensional chaotic time series prediction, IEEE Trans. Neural Netw. Learn. Syst. (2022) 1–12.

[36] Y. Saito, S. Takamichi, H. Saruwatari, Statistical parametric speech synthesis incorporating generative adversarial networks, IEEE/ACM Trans. Audio Lang. Process. 26 (1) (2018) 84–96.

[37] X. Xu, M. Yoneda, Multitask air-quality prediction based on LSTM-autoencoder model, IEEE Trans. Cybern. 51 (5) (2021) 2577–2586.

[38] D. Xu, J.Q. Sheng, P.J.-H. Hu, T.-S. Huang, C.-C. Hsu, A deep learning–based unsupervised method to impute missing values in patient records for improved management of cardiovascular patients, IEEE J. Biomed. Health Inf. 25 (6) (2021) 2260–2272.

[39] A. Rubinsteyn, S. Feldman, Fancyimpute: An imputation library for Python, 2016.