# CGCNImp: a causal graph convolutional network for multivariate time series imputation

**Caizheng Liu** Corresp., 1, 2 , **Guangfan Cui** [1] , **Shenghua Liu** [2]

[1] Department of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing, Beijng, China

[2] Department of Data Science, Institute of Computing Technology, Chinese Academy of Sciences, Beijng, Beijing, China

Corresponding Author: Caizheng Liu
Email address: liucaizheng17b@ict.ac.cn

**Background:** Multivariate time series data generally contains missing values, which can obstacle subsequent analysis and compromise downstream applications. One challenge in this endeavor is the missing values brought about by sensor failure and transmission packet loss. Imputation is the usual remedy in such circumstances. However, in some multivariate time series data, the complex correlation and temporal dependencies, coupled with the non-stationarity of the data, make imputation difficult.

**Mehods:** To address this problem, we propose a novel model for multivariate time series imputation called CGCNImp that considers both correlation and temporal dependency modeling. The correlation dependency module leverages neural Granger causality and a GCN to capture the correlation dependencies among different attributes of the time series data, while the temporal dependency module relies on an attention-driven LSTM and a time lag matrix to learn its dependencies. Missing values and noise are addressed with total variation reconstruction.

**Results:** We conduct thorough empirical analyses on two real-world datasets. Imputation results show that CGCNImp achieves state-of-the-art performance than previous methods.

# CGCNImp : A Causal Graph Convolutional Network for Multivariate Time Series Imputation

**Caizheng Liu[1,2], Guangfan Cui[2], and Shenghua Liu[1]**

[1]**Department of data science,Institute of Computing Technology, Chinese Academy of Sciences,Beijing, China**
[2]**Department of Computer Science and Technology, University of Chinese Academy of Sciences,Beijing,China**

Corresponding author:
Caizheng Liu[1]

Email address: liucaizheng17b@ict.ac.cn

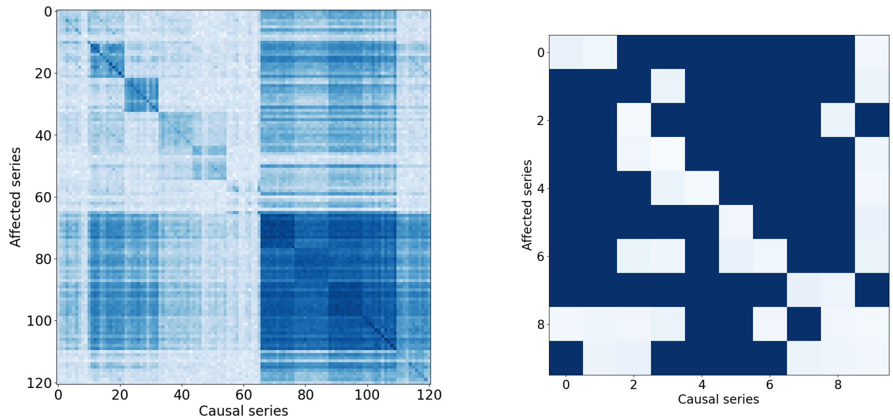
## ABSTRACT

**Background:** Multivariate time series data generally contains missing values, which can obstacle subsequent analysis and compromise downstream applications. One challenge in this endeavor is the missing values brought about by sensor failure and transmission packet loss. Imputation is the usual remedy in such circumstances. However, in some multivariate time series data, the complex correlation and temporal dependencies, coupled with the non-stationarity of the data, make imputation difficult.
**Mehods:** To address this problem, we propose a novel model for multivariate time series imputation called CGCNImp that considers both correlation and temporal dependency modeling. The correlation dependency module leverages neural Granger causality and a GCN to capture the correlation dependencies among different attributes of the time series data, while the temporal dependency module relies on an attention-driven LSTM and a time lag matrix to learn its dependencies. Missing values and noise are addressed with total variation reconstruction.
**Results:** We conduct thorough empirical analyses on two real-world datasets. Imputation results show that CGCNImp achieves state-of-the-art performance than previous methods.

## INTRODUCTION

Multivariate time series data is common to many systems and domains– any data that changes value over time, ~~any~~ data captured by a sensor or measured at intervals ,for example, traffic monitoring (Wang et al., 2017; Zhang et al., 2017), healthcare and patient monitoring (Che et al., 2018; Suo et al., 2019; Liu and Hauskrecht, 2016), IIoT systems, financial marketing (Bauer et al., 2016; Batres-Estrada, 2015) and so on~~, from which the data collected~~ is typically extracted in the form of multivariate time series data. What is also common is missing values and noise brought about by sensor failure, transmission packet loss, human error, and other issues. ~~These missing~~ values will not only destroy the integrity and balance of original data distributions, but also affect the subsequent analysis and application of related scenarios (Cheema, 2014; Berglund et al., 2015). The processing of missing values in time series has become a very important problem. Some researches try to directly model the dataset with missing values (Zheng et al., 2017). However, for every dataset, we need to model them separately. In most cases, imputation is the standard remedy, but imputing with multivariate time series data is not so easy. The complex correlation and temporal dependencies found in some multivariate time series data complicates matters, and the non-stationarity of the data only exacerbates the issue~~, explained as follows:~~

**Attribute correlation dependencies :** In many multivariate time series, it is important to interpret the attribute correlation within time series that naturally arises. Typically, this correlation provides information about the contemporaneous and lagged relationships within and between individual series and how these series interact (Tank et al., 2021, 2018). Fig. 1 (a) illustrates the causal relationship graph with the KDD time series which collects air quality and weather data. In this data, there are 121 variables in total ~~being~~ 11 different locations, each with 11 different variables. Different attributes for the same places are arranged in adjacent positions. A dark blue element $(i, j)$ means that there is a strong Granger causal effect from variable $i$ to variable $j$. It can be seen that the causal effect is strong along the diagonal of the matrix, which means that there are strong causal effects among different variables at the same location. Several research teams have also demonstrated that many aspects of weather, including temperature, precipitation, air pressure, wind speed, and wind direction, have substantial impacts on the migration of birds and that those impacts are inherently nonlinear (Clairbaux et al., 2019; Bozó et al., 2018). Hence, when attempting to impute missing values, all of these factors must be taken into account and the correlations between all these factors needs to be properly modeled to arrive at an accurate result.




(a) the causal effect matrix of KDD dataset (b) the causal effect of bird migration dataset

**Figure 1.** The causal effect matrix of KDD dataset and bird migration dataset.The $X$ axis indicates attributes. The $Y$ axis indicates attributes. The matrix indicates the causal effect between attributes

**Temporal auto-correlation dependencies :** The evolution of multivariate time series changes dynamically over time and is mainly reflected in auto-correlations and trends (Anghinoni et al., 2021). For example, in bird migration case, factors affecting these correlations can include inadequate food and subsequent starvation, too little energy to travel, bad weather conditions, and others (Visser et al., 2009).

Researchers have proposed various methods of imputing missing values for time series data. The most recent techniques include using the complete data of existing observations to build a model or learning the data distribution and then using that distribution to estimate the missing values. The current models and algorithms with good prediction performance include imputation methods based on ~~machine learning,~~ recurrent neural networks (RNNs) (Che et al., 2018; Suo et al., 2019), and generative adversarial networks (GANs) (Goodfellow et al., 2014). Recently, autoencoders have also been used to impute missing values in multivariate time series data. These represent the current state-of-the-art. For instance, Fortuin et al. (~~Fortuin et al.,~~ 2020) proposed a model based on a deep autoencoder that maps the missing values of multivariate time series data into a continuous low-dimensional hidden space. This framework treats the low-dimensional representations as a Gaussian process but does not specify the goal of learning ~~to generate~~ real samples. Rather, the model simply tries to generate data that is close to a real sample. The result is a set of fuzzy samples. GlowImp (Liu et al., 2022) combines Glow-VAEs and GANs into a generative model that simultaneously learns to encode, generate and compare dataset samples. Although all these systems perform well at their intended task, none consider complex attribute correlations or temporal auto-correlation dependencies.

To fill this gap in the literature, we propose a novel model for multivariate time series imputation called CGCNImp that tackles two challenges. CGCNImp leverages neural Granger causality and a GCN to capture correlation dependencies between the attributes and an attention-driven LSTM plus a time lag matrix to model temporal auto-correlation dependencies and ~~generate~~ the missing values. Last, neighbors with similar values are used to smooth the time series and reduce noise. In summary, our main contributions include:

- A novel model for imputing multivariate time series that considers both attribute correlation and temporal auto-correlation dependencies. The combination of neural Granger causality, an attention mechanism and time lag decay yields satisfactory performance compared to the current methods.

- An imputation technique based on Granger causality and a GCN that captures attribute correlations for higher accuracy. In addition, an attention mechanism and total variation reconstruction automatically recovers latent temporal information.
- We conduct thorough empirical analyses on two real-world datasets. Imputation results show that CGCNImp achieves state-of-the-art performance ~~than~~ previous methods.

**Reproducibility: Our open-sourced code and the data used ~~with the supplement document~~ are available at https://github.com/zhewen166/CGCNImp.**

# RELATED WORK

In recent years, ~~researchers have proposed~~ large body of literature on the imputation of missing value. Due to the limited space, we only describe a few closely related ~~ones~~.

## Statistical ~~Based~~ Methods

Statistical (Little and Rubin, 2019) imputation algorithms impute the missing values with mean value (Kantardzic, 2011), median value (na Edgar and Caroline, 2004), mode value (Donders et al., 2006) ~~and~~ last observed valid value (Amiri and Jensen, 2016), ~~which may impute the missing value by the same value (for example median value) if the missing rate is very high.~~

## Machine Learning Based Methods

Some researchers impute the missing values with ~~M~~achine learning algorithm showing that machine learning based imputation methods are useful for time series imputation. K-Nearest Neighbor (KNN) (Liew et al., 2011) uses pairwise information between the target with missing values and the k nearest reference to impute the missing values. Expectation- Maximization (EM) (Nelwamondo et al., 2007) carry ~~on~~ a multi-step process which predicts the value of the current state and then two estimators refining the predicted values ~~if~~ given state, maximizing a likelihood function. The Matrix Factorization (MF) (C. Li et al., 2015) uses ~~the~~ low rank matrix to estimate the missing value. Tensor Singular Value Decomposition (t-SVD) (Jingfei He and Geng, 2016) initializes the missing values as zeroes. It carries ~~on the~~ SVD decomposition and selects the k most significant columns of V, using a linear combination of these columns to estimate the missing values. Multivariate Imputation by Chained Equations (MICE) (Azur et al., 2011; Buuren and Groothuis-Oudshoorn, 2011) uses a chained equation to fill the missing values. Autoregressive (S. Sridevi et al., 2011) estimates missing values using autoregressive-model. Vector autoregressive imputation method (VAR-IM) (Bashir and Wei, 2018) is based on a vector autoregressive (VAR) model by combining an expectation and minimization algorithm with the prediction error minimization method. Gradient-boosted tree (Friedman, 2020) model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.

## Deep Learning Based Methods

In time series imputation, can be classified into RNN-based methods, VAE-based methods and GAN-based methods.

**RNN-Based methods.** GRU-D (Che et al., 2018) predicts the missing variable by the combination of last observed value, the global mean and the time lag. But, it has drawbacks on general datasets (Che et al., 2018). M-RNN (Yoon et al., 2017) utilizes bi-directional RNN to impute missing values since both previous series and future series of missing values are known. BRITS (Cao et al., 2018) ~~only~~ use the RNN structure to model the time series including Unidirectional Uncorrelated Recurrent Imputation, Bidirectional Uncorrelated Recurrent Imputation and Correlated Recurrent Imputation algorithm. All these models may suffer from ~~bias vanish or exploding problems~~ (Bengio et al., 2015) and error accumulation when encountering the continuous missing values.

**VAE-Based methods.** VAE (Kingma and Welling, 2014) ~~proposed a novel~~ for efficient approximate inference with continuous latent variables. HI-VAE (Nazábal et al., 2020) deals with missing data on Heterogeneous and Incomplete Data. ~~But~~ HI-VAE ~~does~~ not suitable for time series data as it do not exploit temporal information. GP-VAE (Fortuin et al., 2020) combine variational autoencoders and Gaussian processes for time series data. The VAE maps the missing data from the input space into a latent space where the temporal dynamics are modeled by the GP. GlowImp (Liu et al., 2022) combines Glow-VAEs and GANs into a generative model that simultaneously learns to encode, generate and compare dataset samples. All these methods only optimize the lower bound and do not specify the goal of learning to generate real samples.

**GAN-Based methods.** Goodfellow et al. (2014) introduced the generative adversarial networks (GAN), which trains generative deep models via an adversarial process. GAIN (Yoon et al., 2018) has some unique features. The generator receives noise and mask as an input data and the discriminator gets some additional information via a hint vector to ensure that the generator generates samples depending on the true data distribution. But GAIN is not suitable for time series. GRUI-GAN (Luo et al., 2018) proposed a two second stage GAN based. The G tries to generate the realistic time series from the random noise vector z. The D tries to distinguish whether the input data is real data or fake data. The adversarial structure can improve accuracy. But This two-stage training needs a lot more time to train the "best" matched data and seems not stable with a random noise input. E2GAN (Luo et al., 2019) can impute the incomplete time series via end-to-end strategy. This work proposes an encoder-decoder GRUI based structure as the generator which can improve the accuracy and stable when training the model. the discriminator consists a GRUI layer and a fully connected layer working as the encoder. SSGAN (Miao et al., 2021) propose a novel semi-supervised generative adversarial network model, with t a generator, a discriminator, and a classifier to predict missing values in the partially labeled time series data (Che et al., 2018; Cao et al., 2018; Luo et al., 2019).

# METHODOLOGY

## Motivation

In many multivariate time series, it is important to interpret the attribute correlation within time series that naturally arises. Generally, these correlation can be divided into attribute correlation dependencies and temporal auto-correlation dependencies. Hence, our work includes three main considerations: these two types of dependencies plus end-to-end multi-task modeling to properly capture both.

**Attributes correlation dependency** Typically, this correlation provides information about the contemporaneous and lagged relationships within and between individual series and how these series interact (Tank et al., 2021, 2018). For example, in bird migration case, the main attribute dependencies are weather factors such as temperature, air pressure, and wind conditions. All can have a substantial impact on evolution of multivariate time series (Clairbaux et al., 2019; Bozó et al., 2018). These therefore need to be considered if one is to accurately impute any missing values. At the same time, there may be false correlation between some attributes. Hence, determining reasonable causal effects among different attributes is also an important issue. We opted for neural Granger causality (Tank et al., 2021, 2018) to model the correlation dependencies between the variables because it has achieved satisfactory performance on multivariate time series causal inference and it could be easily integrated into the multivariate time series imputation framework.

**Temporal auto-correlation dependency**. The evolution of multivariate time series changes dynamically over time and patterns are quasi-periodical on different scales of years and days (Anghinoni et al., 2021). Additionally, sensor malfunctions and failures, transmission errors, and other factors can mean the recorded time series carries noise (Han and Wang, 2013). Effectively exploiting auto-correlation relationships and eliminating sensor noises is therefore a key consideration.

**Multitask modeling**. Classical time series imputation methods adopt a two-stage modeling approach (Luo et al., 2018; Yoon et al., 2018; Miao et al., 2021). First, they analyze the correlations between multiple sequences and then impute the different sequences separately. However, these two-stage methods can not guarantee the global optimum. In this paper, we aim to establish an end-to-end model for Granger causal analysis and deep-learning-based time series imputation under the same framework, which will hopefully accelerate the imputation process and provide interpretability.

## Preliminary

**Definition 1: Multivariate Time Series.** A multivariate time series $X = \{x_1, x_2, \ldots, x_n\}$ is a sequence with data observed on $n$ timestamps $T = (t_0, t_1, \ldots, t_{n-1})$. The $i-th$ observation $x_i$ contains $d$ attributes $(x_i^1, x_i^2, \ldots, x_i^d)$.

**Example 1: Multivariate Time Series.** We give an example of the multivariate time series $X$ with missing values, / indicates the missing value.

$$X = \begin{bmatrix} 5 & / & / & / & 18 \\ 12 & 32 & 9 & / & 76 \\ 2 & / & 24 & / & 47 \end{bmatrix}$$

**Definition 2: Binary Mask Matrix.** Time series $X$ may contain missing values, and a binary mask vector $\mathbb{R}^{n\times d}$ is introduced to indicate the missing positions, which is defined as:

$$M_i^j = \begin{cases} 0, & \text{if } x_i^j \text{ is null} \\ 1, & \text{otherwise} \end{cases}$$

if the $j$-th attribute of $x_i$ is observed, $M_i^j$ is set to 1. Otherwise, $M_i^j$ is set to 0.

**Example 2: Binary Mask Matrix.** We can thus compute the binary mask matrix according to the multivariate time series X in example 1 which have missing values.

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

**Definition 3: Time Lag Matrix.** In order to record the time lag between current value and last observed value, we introduce the time lag matrix $\delta \in R^{n*d}$. The following formation shows the calculation of the $\delta$.

$$\delta_t^d = \begin{cases} s_t - s_{t-1} + \delta_{t-1}^d & \text{if } t > 0 \text{ and } M_{t-1}^d == 0 \\ s_t - s_{t-1} & \text{if } t > 0 \text{ and } M_{t-1}^d == 1 \\ 0 & \text{if t} == 0 \end{cases}$$

## CGCNImp model

To impute reasonable values in place of the missing values, as shown in Fig. 2, the model contains an attribute correlation dependency module and a temporal auto-correlation dependency module. The correlation dependency module leverages neural Granger causality and a GCN to capture the correlation dependencies between attributes. The output of this module is passed to the temporal dependency module, which combines an attention-driven LSTM with a time lag matrix to generate the missing values. Last, a noise reduction and smoothness module uses neighbors with similar values to smooth the time series and remove much of the noise, while still preserving occasional rapid variations in the original signal. The details of each of these modules and the framework as a whole are discussed in the following sections.

### *Attributes causality modeling*

Determining complex correlation dependencies is a key problem in the process of imputing with multivariate time series data. Here we use the neural Granger causality (Tank et al., 2021, 2018) to model the correlation dependency between the attributes of the multivariate time series. Let $x_t \in R^d$ be a d-dimensional stationary time series and assume we have observed the process at n timestamps $T = (t_0, t_1, \ldots, t_{n-1})$. The basic idea of neural Granger causality is to gauge the extent to which the past activity of one time series is predictive of another time series. Thus, let $h_t \in R^d$ represents the d-dimensional hidden state at time $t$, ~~the~~ represents the historical context of the time series for predicting a component $x_{ti}$. The hidden state at time $t+1$ is updated recursively

$$h_{t+1} = f(x_t, h_t) \tag{1}$$

where $f$ is some nonlinear function that depends on the particular recurrent architecture. We opted for an LSTM to model the recurrent function $f$ due to its effectiveness at modeling complex time dependencies. The standard LSTM model takes the form

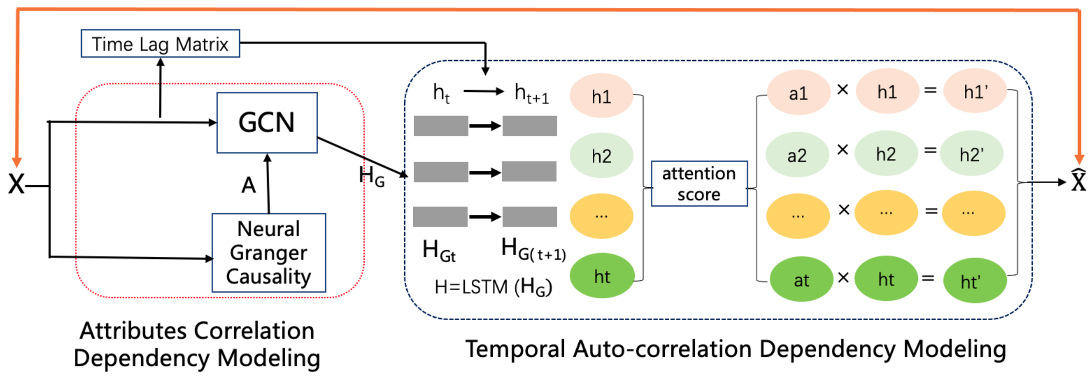



**Figure 2.** The CGCNImp framework for multivariate time series missing value imputing.

$$
\begin{aligned}
f_t &= \sigma(W_f x_t + U_f h_{t-1}) \\
i_t &= \sigma(W_i x_t + U_i h_{t-1}) \\
o_t &= \sigma(W_o x_t + U_o h_{(t-1)}) \\
c_t &= f_t \odot c_{t-1} + i_t \odot tanh(W_c x_t + U_c h_{t-1}) \\
h_t &= o_t \odot tanh(c_t)
\end{aligned}
\tag{2}
$$

where $\odot$ denotes component-wise multiplication and $i_t$, $f_t$, and $o_t$ represent input, forget and output gates, respectively. These control how each component of the state cell $c_t$, is updated and then transferred to the hidden state used for prediction $h_t$. $W_f, W_i, W_o, W_c, U_f, U_i, U_o, U_c$ are the parameters that need to learn by LSTM. The output for series $i$ is given by a linear decoding of the hidden state at time $t$ :

$$x_{ti} = g_i(x_{<t}) + e_{ti} \tag{3}$$

where the dependency of $g_i$ on the full past sequence $x_{<t}$ is due to recursive updates of the hidden state. The LSTM model introduces a second hidden state variable $c_t$, referred to as the cell state, giving the full set of hidden parameter as $(c_t, h_t)$.

In Eq. 2 the set of input maxtrices,

$$W = ((W_f)^T, (W_i)^T, (W_o)^T, (W_c)^T)^T \tag{4}$$

controls how the past time series $x_t$, influences the forget gates, input gates, output gates, and cell updates, and, consequently, the update of the hidden representation. A group lasso penalty across the columns of $W$ can be selected to indicate which Granger series causes series $i$ during estimation. The loss function ~~of~~ for modeling the attribute correlation dependencies is as follows:

$$L_{NG} = \min_{W,U,W_o} \sum_{t=2}^{T} (x_{it} - g_i(x_{<t}))^2 + \lambda \sum_{j=1}^{d} ||W|| \tag{5}$$

where $U = (((U_f)^T, (U_i)^T, (U_o)^T, (U_c)^T)^T)$ . The adjacent matrix A, which is produced by neural granger causality is stated as:

$$A_{ij} = ||W^i_{g_j}||_F^2 \tag{6}$$

***Attributes Correlation Dependency Modeling***

Convolutional neural networks (CNNs) can derive local correlation features but can only be used in Euclidean space. GCNs, however, are semi-supervised models that can handle arbitrary graph-structured data. As such, they have received widespread attention. GCNs can include spectrum and/or spatial domain convolutions. In this study, we use spectrum domain convolutions. In the Fourier domain, spectral convolutions on graphs are defined as the multiplication of a signal $x$ with a filter with a filter $g_\theta : g_\theta * x = U g_\theta (U^T x)$. Here U is the matrix of eigenvectors of the normalized graph Laplacian $L = I_N - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = U \lambda\, U^T$, $U^T x$ is the graph Fourier transform of $x$ ,$A \in R^{d*d}$ is an adjacency matrix and $\lambda$ is diagonal matrix of its eigenvalues. In multivariate time series, $x$ can also be a $X \in R^{n*d}$, where d refers to the number of features and n refers to the time internals. Given the adjacent matrix A which is produced by neural granger causality, GCNs can perform the spectrum convolutional operation with consideration capture the correlation characteristics of graph. The GCN model can be expressed as:

$$H_G = \sigma(\widetilde{W}^{-\frac{1}{2}} \widetilde{A} \widetilde{W}^{-\frac{1}{2}} X \theta^{)} \tag{7}$$

where $\widetilde{A} = A + I_N$ is an adjacent matrix with self-connection structures, $I_N$ is an identity matrix, $\widetilde{W}$ is a degree matrix, $H_G \in R^{n*d}$ is the output of GCN which is the input of the temporal auto-correlation dependency modeling, $\theta$ is the parameter of GCN, and $\sigma(\cdot)$ is an activation function used for nonlinear modeling.

### *Temporal Auto-correlation Dependency Modeling*

Obtaining complex temporal auto-correlation dependencies is another key problem with imputation of multivariate time series data. In particular, sometimes the input decay may not fully capture the missing patterns since not all missingness information can be represented in decayed input values. Due to its effectiveness at modeling complex time dependencies, we choose to model the temporal dependencies using an LSTM (Graves, 2012). However, to properly learn the characteristics of the original incomplete time series dataset, we find that the time lag between two consecutive valid observations is always changing due to the nil values. Further, the time lags between observations are very important since they follow an unknown non-uniform distribution. These changeable time lags remind us that the influence of the past observations should decay with time if a variable has been missing for a while.

Thus, a time decay vector $\alpha$ is introduced to control the influence of the past observations. Each value of $\alpha$ should be greater than 0 and fewer than 1 with the larger the $\delta$, the smaller the decay vector. Hence, the time decay vector $\alpha$ is modeled as a combination of $\delta$:

$$\alpha_t = 1/e^{max(0, W_\alpha \delta_t + b_\alpha)} \tag{8}$$

where $W_\alpha$ and $b_\alpha$ are parameters that need to learn. Once the decay vector has been derived, the hidden state in the LSTM $h_{t-1}$ is updated in an element-wise manner by multiplying the decay vector $\alpha_t$ to fit the decayed influence of the past observations. Thus, the update functions of the LSTM are as follows:

$$\begin{aligned}
h'_{t-1} &= \alpha_t \odot h_{t-1} \\
i_t &= \sigma(W_i[h'_{t-1}; H_{G_t}] + b_i) \\
f_t &= \sigma(W_f[h'_{t-1}; H_{G_t}] + b_f) \\
s_t &= f_t \odot s_{t-1} + i_t \odot tanh(W_s[h'_{t-1}; H_{G_t}] + b_s) \\
o_t &= \sigma(W_o[h'_{t-1}; H_{G_t}] + b_o) \\
h_t &= o_t \odot tanh(s_t)
\end{aligned} \tag{9}$$

where $W_f, W_i, W_o, W_c, b_f, b_i, b_o, b_s$ are the parameters that need to learn by LSTM and $H_G$ is the output of attributes correlation dependency modeling.

Attentive neural networks have recently demonstrated success in a wide range of tasks and, for this reason, we use one here. Let $H_L$ be a matrix consisting of output vectors $H_L = [h_1, h_2, ..., h_n] \in \mathbb{R}^{T \times d}$ that the LSTM layer produced, where n is the time series length. The representation $\beta_{ij}$ of the attention score is formed by a weighted sum of these output vectors:

$$\beta_{ij} = \frac{exp(tanh(W[h_i|h_j]))}{\sum_{k=1}^{T} exp(tanh(W[h_k|h_j]))} \tag{10}$$

$$H^{'} = Atten_L * H_L \tag{11}$$

where the $Atten_L = \begin{bmatrix} \beta_{11}, \beta_{12}, \ldots, \beta_{1T} \\ \beta_{21}, \beta_{22}, \ldots, \beta_{2T} \\ \cdots \\ \beta_{n1}, \beta_{n2}, \ldots, \beta_{nT} \end{bmatrix}$ is the attention score.

In Eq.11, We reconstruct the missing value by some linear transformation of the hidden state $H^{'}$ at time t. Hence the reconstruction loss is formulated as:

$$L_{reg} = \sum_{x \in D} \|x \otimes m - \hat{x} \otimes m\|_2 \tag{12}$$

x represents the input multivariate time series data, $\hat{x}$ represents the imputed multivariate time series data and m means the masking matrix. The expression in Eq. 12 is the masked reconstruction loss that calculates the squared errors between the original observed data x and the imputed sample. Here, it should be emphasized that when calculating the loss, we only calculate the observed data as previously described in (Cao et al., 2018; Luo et al., 2018, 2019; Liu et al., 2022).

### *Noise Reduction and Smoothness Imputation*

In the past, reconstructions were performed directly, which ignores the noise in the actual sampling process. However, in real-world multivariate time series data, when time series are collected the observations may be contaminated by various types of error or noise. Hence, these imputation values may be unreliable. To ensure the reliability of the imputation results, a total variation reconstruction regularization term is applied to the reconstruction results. The method is based on a smoothing function where neighbors with similar values are used to smooth the time series. When applied to time series data, abrupt changes in trend, spikes, dips and the like can all be fully preserved. This regularization term is formulated as follows:

$$\sum_{j=1}^{M} \sum_{i=1}^{N-1} |\hat{x}_{i+1}^{j} - \hat{x}_{i}^{j}| \tag{13}$$

where M is the number of time series, that is, the number of variables, and N is the length of each time series. Compared to a two-norm smoothing constraint, this total variation reconstruction term can ensure smoothness without losing the dynamic performance of the time series (Boyd and Vandenberghe, 2004) .

Eq.13 applies this term to the reconstruction results. As a result, noise in the original data is reduced and completion accuracy is improved. The reconstruction loss is formulated as:

$$L_{SL} = \sum_{j=1}^{M} \sum_{i=1}^{N} |\hat{x}_{i+1}^{j} - \hat{x}_{i}^{j}| \tag{14}$$

The total object function of our model is:

$$L_{loss} = \alpha * L_{NG} + \beta * L_{reg} + \theta * L_{SL} \tag{15}$$

where $\alpha$,$\beta$,$\theta$ indicate the weight among different part of the total loss. We optimize Eq. 15 using proximal gradient descent with line search.

## EXPERIMENT

To ~~accurately~~ verify and measure the performance of the proposed CGCNImp framework, we compare its performance at imputation with multiple time series against several other contemporary methods. The selected datasets used in the evaluations were two real-world bird migration datasets focusing on migratory patterns in China – Anser albifrons and Anser fabalis – as well as the KDD 2018 CUP Dataset.

## Dataset Description

### *KDD CUP 2018 Dataset*

The KDD dataset comes from the KDD CUP Challenge 2018 [1]. The dataset which is is a public meteorologic dataset is about 15% missing values. It was hourly collected between 2017/1/20 to 2018/1/30 of Beijing collecting air quality and weather data . Each record contains 12 attributes,for example CO, weather, temperature etc. In our experiment, We select 11 common features for our experiments 12 attributes as the previous method did. We split this dataset for every hour. For every 48 hours , we randomly drop p percent of the dataset,and then we impute these time series with different models and calculate the imputation accuracy by RMSE and MAE where $p \in \{10,20,30,40,50,60,70,80,90\}$.

### *Bird Migration Dataset in China*

The Birds Migration Dataset collects migration trace data which comes from the project Strategic Priority Research Program of Chinese Academy of Sciences which. The dataset was hourly collected between 2017/12/30 to 2018/5/10 of Anser fabalis and Anser albifrons. Each record contains 13 attributes which are longitude, latitude, speed height,speed velocity, heading, temperature etc. The dataset is about 10% missing values. We select 10 common features contains longitude, latitude, speed height,speed velocity, heading, temperature etc. for our experiments. We split this dataset for 5 minutes time series, and for every 5 minutes, we randomly drop p percent of the dataset,and then we impute these time series with different models and calculate the imputation accuracy by RMSE and MAE between original values and imputed values where $p \in \{10,20,30,40,50,60,70,80,90\}$.

## Comparison Methods and Evaluation Metrics

We compare our methods to eight current imputation methods as previously described in (Liu et al., 2022). A brief description of each follows.

- Statistical imputation methods (Rubinsteyn and Feldman, 2016), where we simply impute the missing values with zero, mean , median.
- KNN (Liew et al., 2011), which the missing data is imputed as the weighted average of k neighbors by using a k-nearest neighbor algorithm to find neighboring data.
- MF (C. Li et al., 2015), which fills the missing values through factorizing an incomplete matrix into low-rank matrices .
- SVD (Jingfei He and Geng, 2016), which uses iterative singular value decomposition for matrix imputation to impute the missing values.
- GP-VAE (Fortuin et al., 2020), a method that combines ideas from VAEs and Gaussian processes to capture temporal dynamics for time series imputation.
- BRITS (Cao et al., 2018), one of methods that include Unidirectional Uncorrelated Recurrent Imputation,Bidirectional Uncorrelated Recurrent Imputation and Correlated Recurrent Imputation algorithm to impute the missing values.
- GRUI (Luo et al., 2018), which is a two-stage GAN based method that use the generator and discriminator to impute missing values.
- E2E-GAN (Luo et al., 2019). It relies on an end-to-end GAN network that proposes an encoder-decoder GRUI based structure and is one of the state-of-the-art methods.

To evaluate the performance of our methods, we use two metrics to the compare and analyze with the results of previous methods.

(1) RMSE (Root Mean Squared Error) refers to the mean value of the square root of the error between the predicted value and the true value. This kind of measurement method is very popular, it can better describe the data, and it is a quantitative weighing method. The calculation formula is as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x-\hat{x})^2}$$

[1] KDD CUP. Available on: http://www.kdd.org/kdd2018/, 2018.

(2) MAE(Mean Absolute Error) is the average of the absolute value of the error between the observed value and the real value. it is used to describe the error between the predicted value and the real value. The formulation is as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|x-\hat{x}|$$

## Implementations Details

All the experimental results are obtained under the same hardware and software environment. The hardware is Intel i7 9700k, 48GB memory, NVIDIA GTX 1080 8GB. And the deep learning framework is PyTorch1.7 and TensorFlow1.15.0.

To maintain the same experiment environment as the contemporary method, the dataset was split the datasets into two parts which first part with 80% of the records is used for the training set and the remaining 20% is used for the test set. All values are normalized within the range of 0 to 1. For training process, 10% of the data of the training set was randomly dropped. When testing dataset, we drop the data with different drop-rate between 10% and 90%, tested each method at a range of levels of missing data between 10% and 90%.

## Performance Analysis

The results with the KDD, Anser albifrons and Anser fabalis datasets at a missing value ratio of 10% appear in Table 1. Here, CGCNImp yields significantly fewer errors than the other methods in terms of RMSE and MAE, demonstrating that our method is better than other methods.

| dataset | KDD dataset | | Anser albifrons dataset | | Anser fabalis dataset | |
|---|---|---|---|---|---|---|
| Method | RMSE | MAE | RMSE | MAE | RMSE | MAE |
| Zero | 1.081 | 1.041 | 1.088 | 1.047 | 1.089 | 1.054 |
| Mean | 1.063 | 1.035 | 1.033 | 1.025 | 1.043 | 1.035 |
| Random | 1.821 | 1.637 | 1.802 | 1.431 | 1.721 | 1.677 |
| Median | 1.009 | 0.994 | 1.109 | 1.042 | 1.001 | 0.998 |
| KNN | 0.803 | 0.724 | 0.758 | 0.714 | 0.824 | 0.817 |
| MF | 0.784 | 0.627 | 0.643 | 0.626 | 0.663 | 0.646 |
| SVD | 1.043 | 0.966 | 1.253 | 1.051 | 1.129 | 1.011 |
| GP-VAE | 0.597 | 0.486 | 0.693 | 0.572 | 0.534 | 0.375 |
| BRITS | 0.156 | 0.148 | 0.159 | 0.124 | 0.137 | 0.078 |
| GRUI | 0.149 | 0.102 | 0.152 | 0.113 | 0.138 | 0.086 |
| E2E-GAN | 0.133 | 0.074 | 0.139 | 0.081 | 0.116 | 0.066 |
| Ours | **0.114** | **0.062** | **0.128** | **0.072** | **0.107** | **0.059** |

**Table 1.** The RMSE and MAE results of the CGCNImp and other methods on two datasets(lower is better).

Generally , the higher the proportion of missing data, the more difficult it is to impute the missing value. However, the proportion of missing data is often uncertain. The prediction ability of the model is very important with different missing ratio. To assess the frameworks with different levels of missing data, we then conduct the same experiment with the BRITS,GRUI, E2EGAN and CGCNImp , varying the ratios of missing values from 10% to 90% in steps of 10% as previously described in (Liu et al., 2022). The results are shown in Table 2 and Table 3. Again, our methods return the fewest errors.

Fig. 3 , Fig. 4 and Fig. 5 show the imputation results from the KDD datasets for the Tongzhou , Mentougou and Miyun districts, respectively. The blue dots are the ground truth time series and the red curve shows the imputed values. As illustrated, CGCNImp captures the evolution trend and imputes the missing values quite well. Further, it capture the potential probability density distribution of the multivariate time series and makes full use of the interactive information available.

| | | missing rate (%) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 |
| KDD | BRITS | 0.1484 | 0.1587 | 0.1722 | 0.1924 | 0.2076 | 0.2391 | 0.2624 | 0.3076 | 0.3591 |
| | GRUI | 0.1026 | 0.1397 | 0.1522 | 0.1684 | 0.1876 | 0.1971 | 0.2341 | 0.2863 | 0.3198 |
| | E2EGAN | 0.0747 | 0.1087 | 0.1292 | 0.1428 | 0.1576 | 0.1796 | 0.1981 | 0.2076 | 0.2590 |
| | Ours | **0.0624** | **0.0721** | **0.0814** | **0.0959** | **0.1109** | **0.1238** | **0.1397** | **0.1564** | **0.1745** |
| albifrons | BRITS | 0.1242 | 0.1289 | 0.1331 | 0.1446 | 0.1576 | 0.1891 | 0.2021 | 0.2386 | 0.2896 |
| | GRUI | 0.1133 | 0.1197 | 0.1202 | 0.1377 | 0.1478 | 0.1691 | 0.1723 | 0.1976 | 0.2259 |
| | E2EGAN | 0.0815 | 0.0967 | 0.1026 | 0.1124 | 0.1389 | 0.1582 | 0.1648 | 0.1870 | 0.1993 |
| | Ours | **0.0721** | **0.0785** | **0.0923** | **0.0992** | **0.1152** | **0.1346** | **0.1537** | **0.1714** | **0.2014** |
| fabalis | BRITS | 0.0782 | 0.1371 | 0.1462 | 0.1639 | 0.1978 | 0.2123 | 0.2694 | 0.2971 | 0.3791 |
| | GRUI | 0.0863 | 0.1007 | 0.1252 | 0.1504 | 0.1771 | 0.1925 | 0.2492 | 0.3084 | 0.3271 |
| | E2EGAN | 0.0661 | 0.0787 | 0.0928 | 0.1027 | 0.1285 | 0.1382 | 0.1537 | 0.2004 | 0.2596 |
| | Ours | **0.0591** | **0.0688** | **0.0815** | **0.0928** | **0.1099** | **0.1257** | **0.1467** | **0.1776** | **0.2189** |

**Table 2.** The MAE results of the ~~CGCNImp~~ methods on two datasets with different missing rate (lower is better).

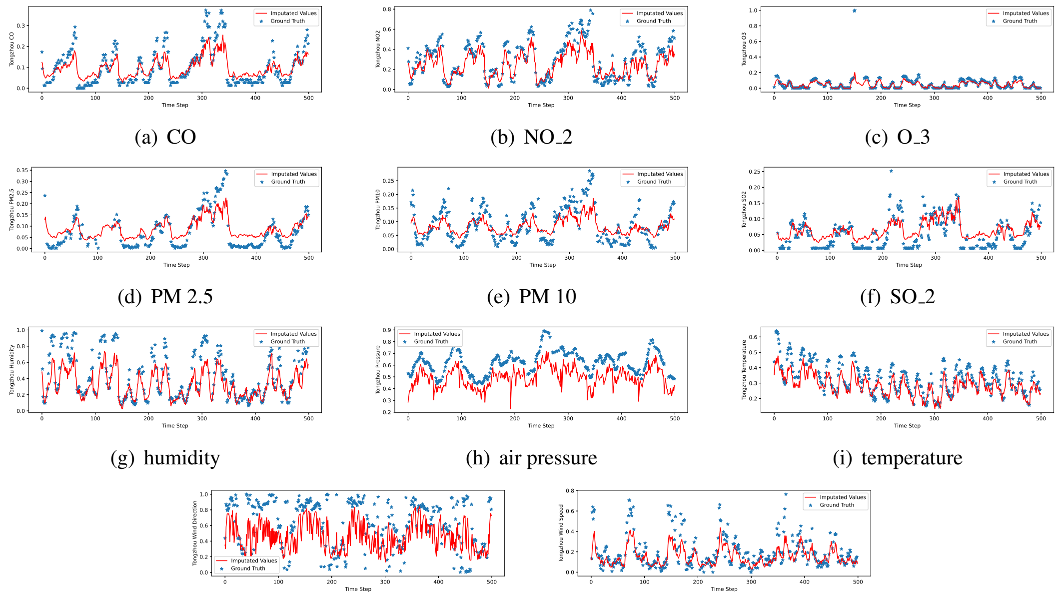
| | | missing rate (%) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 |
| KDD | BRITS | 0.1561 | 0.1721 | 0.1928 | 0.2120 | 0.2571 | 0.2980 | 0.3284 | 0.3625 | 0.3912 |
| | GRUI | 0.1493 | 0.1527 | 0.1702 | 0.1937 | 0.2098 | 0.2541 | 0.2824 | 0.3051 | 0.3361 |
| | E2EGAN | 0.1336 | 0.1457 | 0.1601 | 0.1778 | 0.1926 | 0.2235 | 0.2574 | 0.2808 | 0.3031 |
| | Ours | **0.1142** | **0.1279** | **0.1402** | **0.1610** | **0.1803** | **0.2026** | **0.2263** | **0.2509** | **0.2776** |
| albifrons | BRITS | 0.1596 | 0.1706 | 0.1931 | 0.2126 | 0.2398 | 0.2571 | 0.2964 | 0.3351 | 0.3686 |
| | GRUI | 0.1394 | 0.1562 | 0.1799 | 0.1971 | 0.2205 | 0.2483 | 0.2670 | 0.2995 | 0.3297 |
| | E2EGAN | 0.1289 | 0.1358 | 0.1572 | 0.1704 | 0.1976 | 0.2371 | 0.2480 | 0.2746 | 0.3098 |
| | Ours | **0.1287** | **0.1394** | **0.1589** | **0.1679** | **0.1902** | **0.2163** | **0.2389** | **0.2590** | **0.2921** |
| fabalis | BRITS | 0.1372 | 0.1451 | 0.1680 | 0.1901 | 0.2273 | 0.2398 | 0.2647 | 0.3004 | 0.3469 |
| | GRUI | 0.1381 | 0.1483 | 0.1761 | 0.2007 | 0.2492 | 0.2703 | 0.2906 | 0.3209 | 0.3501 |
| | E2EGAN | 0.1160 | 0.1246 | 0.1508 | 0.1688 | 0.1898 | 0.2103 | 0.2562 | 0.2953 | 0.3391 |
| | Ours | **0.1076** | **0.1242** | **0.1444** | **0.1588** | **0.1829** | **0.2059** | **0.2323** | **0.2713** | **0.3205** |

**Table 3.** The RMSE results of the ~~CGCNImp~~ methods on two datasets with different missing rate (lower is better).

## Ablation study

An ablation study is designed to assess the contribution of the attribute causality discovery and the noise reduction and smoothness imputation. This comprised three tests: the first with no ablation; the second where we simply removed the noise reduction and smoothness module and set $\beta$ to 0 in Eq. 15; plus a third where we simply removed the noise reduction and smoothness module and set $\alpha$ to 0 in in Eq. 15. All tests are conducted with a range of missing value ratios. Table 4 and Table 5 show the results. What we found with the Anser bird migration data was that, at a missing rate lower than 40%, removing either the noise reduction and smoothness module or the neural Granger causality gives fewer errors. However, at higher missing rates, the tests with both modules returned substantially fewer errors. This verifies the contribution of both modules to the framework. With the KDD data, CGCNImp in full returned substantially fewer errors, again supporting the contribution of both these modules.

Fig. 1 (a) illustrates the causal relationship graph with the KDD time series. In this data, there are 121 variables in total ~~being~~ 11 different locations, each with 11 different variables. Different attributes for the same places are arranged in adjacent positions. A dark blue element $(i, j)$ means that there is a strong Granger causal effect from variable $i$ to variable $j$. It can be seen that the causal effect is strong along the diagonal of the matrix, which means that there are strong causal effects among different variables at the same location. Furthermore, there are also strong causal effects between different locations, such as




(a) CO (b) NO_2 (c) O_3

(d) PM 2.5 (e) PM 10 (f) SO_2

(g) humidity (h) air pressure (i) temperature

(j) wind direction (k) wind speed

**Figure 3.** The ground true(blue) and the imputed values(red) in Tongzhou,Beijing of KDD dataset.The $X$ axis indicates time step. The $Y$ axis indicates imputed values.

variables 66 to 110.

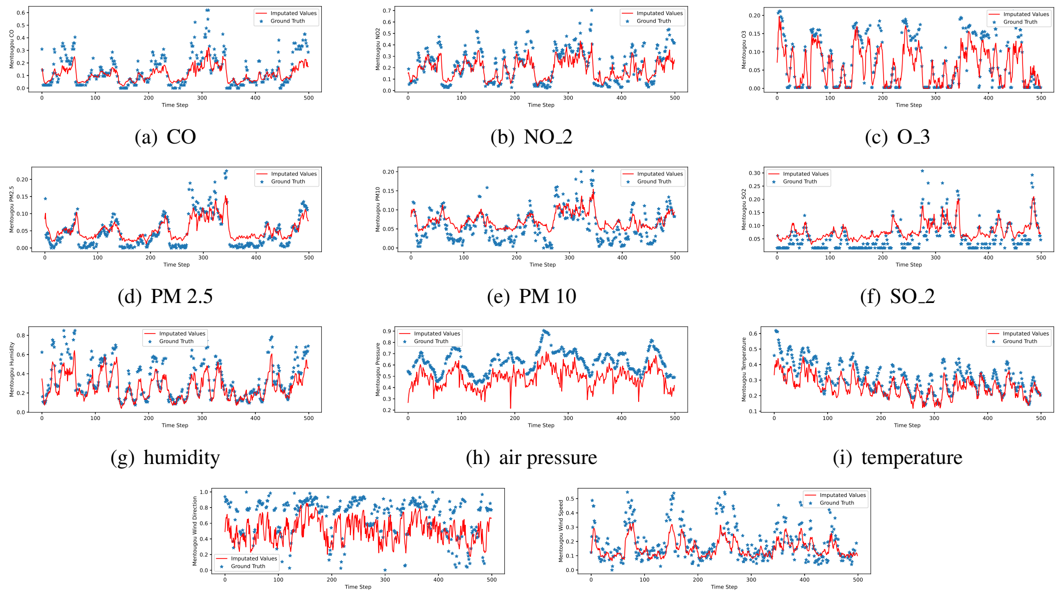
There are 9 attributes in the bird migration dataset. Fig. 1 (b) shows the Granger causal matrix derived from the neural Granger causality analysis. It should be noted that the Granger causality should be a one-way relationship, which means that, theoretically, we need to eliminate conflicting edges in the causal graph. However, in practice, the causal graph is derived from the neural Granger causality analysis and the edge indicate there are strong prediction benefit between variables. Therefore, we kept the conflicting edge and placed them into the GCN network for better performance.

## CASE STUDY : Bird migration route analysis

Fig. 6 , Fig. 7 and Fig. 8 show the imputation results of Anser fabalis birds migration routes. What we can see is that the imputed data shows some important wild reserves not seen with the original data. According to the list of wetlands of international importance in China, for example, Fig. 6 (b) shows the ground truth time series with missing values. This time, CGCNImp imputed the location of Binzhou Seashell Island and the Wetland National Nature Reserve not shown in Fig. 6 (a) showing that the bird migration trajectory could be recovered by our methods.

| | | missing rate (%) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 |
| KDD | $\theta = 0$ | 0.1468 | 0.1607 | 0.1713 | 0.1890 | 0.2071 | 0.2251 | 0.2449 | 0.2634 | 0.2845 |
| | $\alpha = 0$ | 0.1278 | 0.1445 | 0.1607 | 0.1788 | 0.1982 | 0.2173 | 0.2372 | 0.2611 | 0.2826 |
| | no ablation | **0.1142** | **0.1279** | **0.1402** | **0.1610** | **0.1803** | **0.2026** | **0.2263** | **0.2509** | **0.2776** |
| albifrons | $\theta = 0$ | 0.1240 | 0.1329 | 0.1567 | 0.1726 | 0.1918 | 0.2143 | 0.2416 | 0.2694 | 0.3112 |
| | $\alpha = 0$ | **0.1180** | **0.1301** | **0.1473** | 0.1761 | 0.1873 | 0.2098 | 0.2304 | 0.2639 | 0.3047 |
| | no ablation | 0.1287 | 0.1394 | 0.1589 | **0.1679** | **0.1802** | **0.2063** | **0.2289** | **0.2590** | **0.2921** |
| fabalis | $\theta = 0$ | 0.1156 | 0.1296 | **0.1373** | **0.1582** | 0.1827 | 0.2092 | 0.2437 | 0.2833 | 0.3276 |
| | $\alpha = 0$ | 0.1169 | 0.1294 | 0.1393 | 0.1617 | **0.1781** | 0.2132 | 0.2359 | 0.2781 | 0.3230 |
| | no ablation | **0.1076** | **0.1242** | 0.1444 | 0.1588 | 0.1829 | **0.2059** | **0.2323** | **0.2713** | **0.3205** |

**Table 4.** The ablation study RMSE results of the CGCNImp methods on two datasets(lower is better).



(a) CO (b) NO_2 (c) O_3
(d) PM 2.5 (e) PM 10 (f) SO_2
(g) humidity (h) air pressure (i) temperature
(j) wind direction (k) wind speed

**Figure 4.** The ground true(blue) and the imputed values(red) in Mentougou,Beijing of KDD dataset.The $X$ axis indicates time step. The $Y$ axis indicates imputed values.

| | | missing rate (%) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 |
| KDD | $\theta = 0$ | 0.0801 | 0.0925 | 0.1028 | 0.1155 | 0.1289 | 0.1416 | 0.1552 | 0.1688 | 0.1833 |
| | $\alpha = 0$ 0.0717 | 0.0835 | 0.0957 | 0.1090 | 0.1224 | 0.1356 | 0.1492 | 0.1658 | 0.1806 | 0.1866 |
| | no ablation | **0.0624** | **0.0721** | **0.0814** | **0.0959** | **0.1109** | **0.1238** | **0.1397** | **0.1564** | **0.1745** |
| albifrons | $\theta = 0$ | 0.0659 | 0.0739 | 0.0891 | 0.1012 | 0.1163 | 0.1327 | 0.1528 | 0.1764 | 0.2118 |
| | $\alpha = 0$ | **0.0638** | **0.0722** | **0.0833** | 0.1001 | 0.1112 | 0.1284 | 0.1457 | 0.1723 | 0.2083 |
| | no ablation | 0.0721 | 0.0785 | 0.0923 | **0.0992** | **0.1052** | **0.1246** | **0.1437** | **0.1714** | **0.2014** |
| fabalis | $\theta = 0$ | 0.0618 | 0.0721 | 0.0799 | 0.0935 | 0.1106 | 0.1298 | 0.1569 | 0.1879 | 0.2265 |
| | $\alpha = 0$ | 0.0633 | 0.0709 | **0.0795** | 0.0947 | 0.1079 | 0.1315 | 0.1497 | 0.1825 | 0.2218 |
| | no ablation | **0.0591** | **0.0688** | 0.0815 | **0.0928** | **0.1069** | **0.1257** | **0.1467** | **0.1776** | **0.2189** |

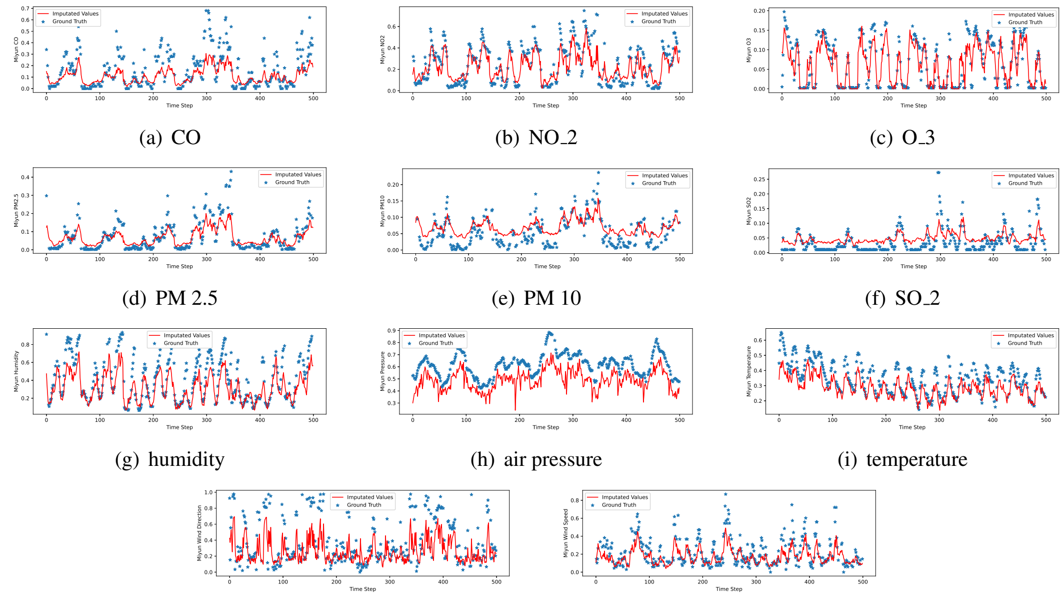
**Table 5.** The ablation study MAE results of the CGCNImp methods on two datasets(lower is better).

Fig. 7 (b) ~~which is the ground truth time series with missing values,~~ CGCNImp method imputed the location of Wanfoshan Nature Reserve which is not noticeable in the original data on its own (Fig. 7 (a)). Wanfoshan is now a national forest park, a national nature reserve, and a national geological park which is an important location for bird migration.

Likewise, Fig. 8 (b) shows the imputed location of the Momoge National Nature Reserve (Cui et al., 2021) not showed in Fig. 8 (a).

# CONCLUSION

In this paper, we present a novel imputation model, called CGCNImp, that is specifically designed to imputation of multivariate time series data. CGCNImp considers both attribute correlation and temporal auto-correlation dependencies. Correlation dependencies are captured through neural Granger causality and a GCN, while an attention-driven LSTM plus a time lag matrix captures the temporal dependencies and generates the missing values. Last, neighbors with similar values are used to smooth the time series and reduce noise. Imputation results show that CGCNImp achieves state-of-the-art performance ~~than~~ previous methods. We will explore our model for missing-not-at-random data and we will conduct



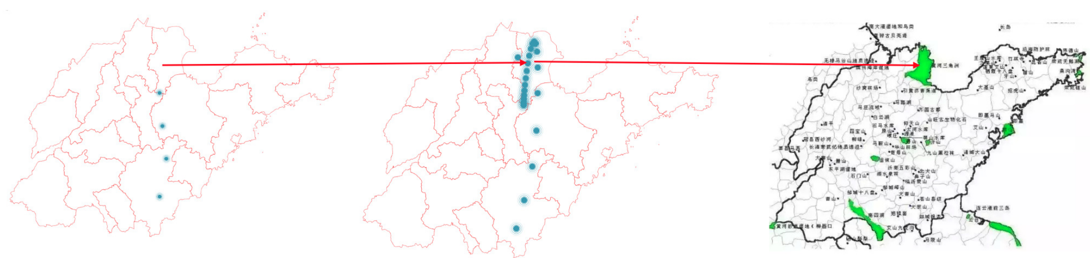
(a) CO (b) NO_2 (c) O_3

(d) PM 2.5 (e) PM 10 (f) SO_2

(g) humidity (h) air pressure (i) temperature

(j) wind direction (k) wind speed

**Figure 5.** The ground true(blue) and the imputed values(red) in Miyun,Beijing of KDD dataset.The $X$ axis indicates time step. The $Y$ axis indicates imputed values.



(a) Anser fabalis 01 before imputation (b) Anser fabalis 01 after imputation (c) The Nature Reserve Zone

**Figure 6.** Anser fabalis dataset imputation.

theoretical analysis of our model for missing values in ~~the further works~~.

## REFERENCES


Amiri, M. and Jensen, R. (2016). Missing data imputation using fuzzy-rough methods. *Neurocomputing*, 205:152–164.

Anghinoni, L., Vega-Oliveros, D., Silva, T., and Zhao, L. (2021). Time series pattern identification by hierarchical community detection. *The European Physical Journal Special Topics*, 230.

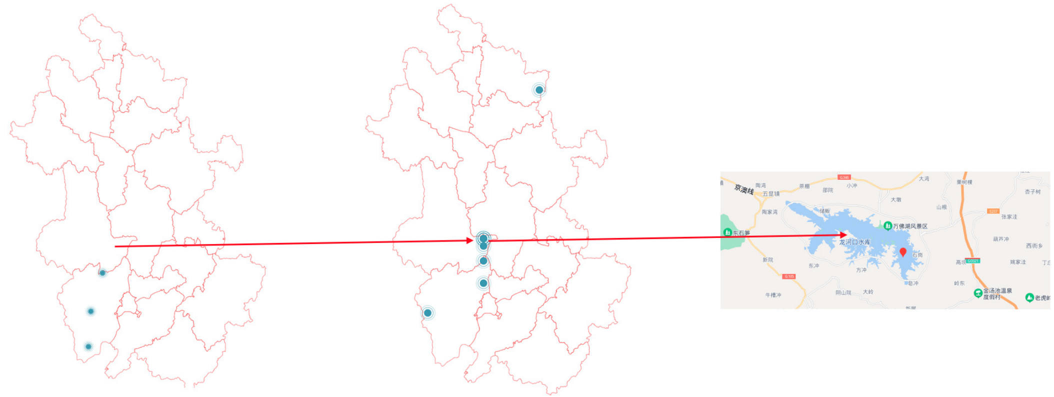
Azur, M. J., Stuart, E. A., Frangakis, C., and Leaf, P. J. (2011). Multiple imputation by chained equations: what is it and how does it work? *International Journal of Methods in Psychiatric Research*, 20(1):40–49.

Bashir, F. and Wei, H.-L. (2018). Handling missing data in multivariate time series using a vector autoregressive model-imputation (VAR-IM) algorithm. *Neurocomputing*, 276:23–30.

Batres-Estrada, B. (2015). Deep learning for multivariate financial time series.

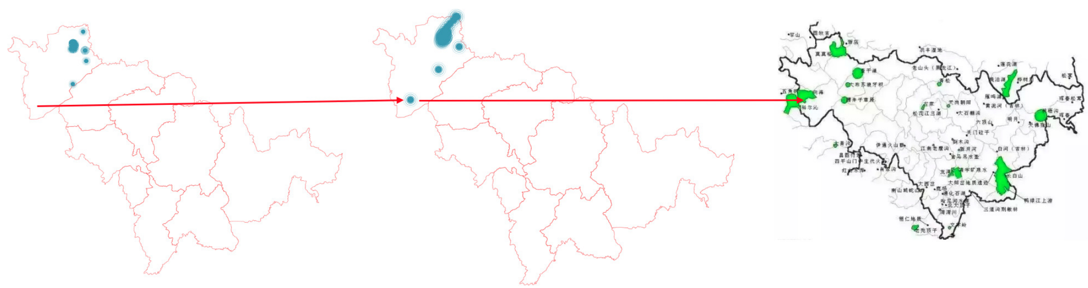
Bauer, S., Schölkopf, B., and Peters, J. (2016). The arrow of time in multivariate time series. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 2043–2051. JMLR.org.

Bengio, S., Vinyals, O., Jaitly, N., and Shazeer, N. (2015). Scheduled sampling for sequence prediction with recurrent neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, pages 1171–1179.



(a) Anser fabalis 03 before imputation (b) Anser fabalis 03 after imputation (c) The Nature Reserve Zone

**Figure 7.** Anser fabalis dataset imputation.



(a) Anser fabalis 02 before imputation (b) Anser fabalis 02 after imputation (c) The Nature Reserve Zone

**Figure 8.** Anser fabalis dataset imputation.


Berglund, M., Raiko, T., Honkala, M., Kärkkäinen, L., Vetek, A., and Karhunen, J. (2015). Bidirectional recurrent neural networks as generative models. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 856–864, Cambridge, MA, USA. MIT Press.

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Bozó, L., Csörgö, T., and Heim, W. (2018). Weather conditions affect spring and autumn migration of siberian leaf warblers. *Avian Research*, 9.

Buuren, S. and Groothuis-Oudshoorn, C. (2011). Mice: multivariate imputation by chained equations in r. *Journal of Statistical Software*, 45.

C. Li, W. Su, T. G. McKenzie, F. Hsu, S. Lin, J. Y. Hsu, and P. B. Gibbons (2015). Recommending missing sensor values. In *2015 IEEE International Conference on Big Data (Big Data)*, pages 381–390.

Cao, W., Wang, D., Li, J., Zhou, H., Li, L., and Li, Y. (2018). Brits: bidirectional recurrent imputation for time series. In *Advances in Neural Information Processing Systems*, volume 31.

Che, Z., Purushotham, S., Cho, K., Sontag, D., and Liu, Y. (2018). Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, 8(1):6085.

Cheema, J. R. (2014). A review of missing data handling methods in education research. *Review of Educational Research*, 84(4):487–508.

Clairbaux, M., Fort, J., Mathewson, P., Porter, W., Strøm, H., and Grémillet, D. (2019). Climate change could overturn bird migration: Transarctic flights and high-latitude residency in a sea ice free arctic. *Scientific Reports*, 9.

Cui, G., Liu, Y., and S, T. (2021). Analysis of the causes of wetland landscape patterns and hydrological connectivity changes in momoge national nature reserve based on the google earth engine platform. *Arabian Journal of Geosciences*, 14(3).

Donders, A. R. T., van der Heijden, G. J., Stijnen, T., and Moons, K. G. (2006). Review: a gentle introduction to imputation of missing values. *Journal of Clinical Epidemiology*, 59(10):1087–1091.

Fortuin, V., Baranchuk, D., Raetsch, G., and Mandt, S. (2020). Gp-vae: Deep probabilistic time series imputation. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108, pages 1651–1661.

Friedman, J. H. (2020). Contrast trees and distribution boosting. *Proceedings of the National Academy of Sciences*, 117(35):21175–21184.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, pages 2672–2680.

Graves, A. (2012). Supervised sequence labelling. *Springer Berlin Heidelberg*.

Han, M. and Wang, X. (2013). Robust neural predictor for noisy chaotic time series prediction. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–5.

Jingfei He, Guiling Sun, Y. Z. and Geng, T. (2016). Data recovery in heterogeneous wireless sensor networks based on low-rank tensors. In *2016 IEEE Symposium on Computers and Communication (ISCC)*, pages 616–620.

Kantardzic, M. (2011). *Data mining: concepts, models, methods, and algorithms*. John Wiley & Sons.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Liew, A. W.-C., Law, N.-F., and Yan, H. (2011). Missing value imputation for gene expression data: computational techniques to recover missing data from available information. *Briefings in Bioinformatics*, 12(5):498–513.

Little, R. J. and Rubin, D. B. (2019). *Statistical analysis with missing data*, volume 793. John Wiley & Sons.

Liu, C., Zhou, H., Sun, Z., and Cui, G. (2022). Glowimp: Combining glow and gan for multivariate time series imputation. In *Algorithms and Architectures for Parallel Processing*, pages 50–64, Cham. Springer International Publishing.

Liu, Z. and Hauskrecht, M. (2016). Learning linear dynamical systems from multivariate time series: A matrix factorization based framework. *Proceedings of the SIAM International Conference on Data Mining. SIAM International Conference on Data Mining*, 2016:810–818.

Luo, Y., Cai, X., Zhang, Y., Xu, J., and Yuan, X. (2018). Multivariate time series imputation with generative adversarial networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 1603–1614.

Luo, Y., Zhang, Y., Cai, X., and Yuan, X. (2019). E²GAN: end-to-end generative adversarial network for multivariate time series imputation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 3094–3100.

Miao, X., Wu, Y., Wang, J., Gao, Y., Mao, X., and Yin, J. (2021). Generative semi-supervised learning for multivariate time series imputation. In *Proceedings of the AAAI Conference on Artificial Intelligence,35(10)*, pages 8983–8991.

na Edgar, A. and Caroline, R. (2004). The treatment of missing values and its effect on classifier accuracy. In *Classification, Clustering, and Data Mining Applications*, pages 639–647.

Nazábal, A., Olmos, P. M., Ghahramani, Z., and Valera, I. (2020). Handling incomplete heterogeneous data using vaes. *Pattern Recognition*, 107:107501.

Nelwamondo, F. V., Mohamed, S., and Marwala, T. (2007). Missing data: a comparison of neural network and expectation maximization techniques. *Current Science*, 93(11):1514–1521.

Rubinsteyn, A. and Feldman, S. (2016). fancyimpute: an imputation library for python.

S. Sridevi, S. Rajaram, C. Parthiban, S. SibiArasan, and C. Swadhikar (2011). Imputation for the analysis of missing values and prediction of time series data. In *2011 International Conference on Recent Trends in Information Technology (ICRTIT)*, pages 1158–1163.

Suo, Q., Yao, L., Xun, G., Sun, J., and Zhang, A. (2019). Recurrent imputation for multivariate time series with missing values. In *2019 IEEE International Conference on Healthcare Informatics (ICHI)*.

Tank, A., Covert, I., Foti, N., Shojaie, A., and Fox, E. (2018). Neural granger causality for nonlinear time series.

Tank, A., Covert, I., Foti, N., Shojaie, A., and Fox, E. B. (2021). Neural granger causality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–1.

Visser, M., PERDECK, A., BALEN, J., and Both, C. (2009). Climate change leads to decreasing bird migration distance. *Global Change Biology*, 15:1859 – 1865.

Wang, D., Cao, W., Li, J., and Ye, J. (2017). Deepsd: Supply-demand prediction for online car-hailing services using deep neural networks. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pages 243–254.

Yoon, J., Jordon, J., and van der Schaar, M. (2018). Gain: missing data imputation using generative adversarial nets. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 5689–5698.

Yoon, J., Zame, W. R., and van der Schaar, M. (2017). Multi-directional recurrent neural networks: A novel method for estimating missing data. In *Time Series Workshop in International Conference on Machine Learning*.

Zhang, J., Zheng, Y., and Qi, D. (2017). Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 1655–1661. AAAI Press.

Zheng, K., Gao, J., Ngiam, K. Y., Ooi, B. C., and Yip, W. L. J. (2017). Resolving the bias in electronic medical records. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 2171–2180, New York, NY, USA. Association for Computing Machinery.