ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230625

Time-Series Data Imputation via Realistic Masking-Guided Tri-Attention Bi-GRU

Zhipeng Zhang[1], Yiqun Zhang[1;*], An Zeng[1;*], Dan Pan[2], Yuzhu Ji[1], Zhipeng Zhang[1] and Jing Lin[1]

[1]Guangdong University of Technology, Guangzhou, China
[2]Guangdong Polytechnic Normal University, Guangzhou, China

Abstract. Time series data with missing values are ubiquitous in real applications due to various unforeseen faults during data generation, storage, and transmission. Time-Series Data Imputation (TSDI) is thus crucial to many temporal data analysis tasks. However, existing works usually consider only one of the following two issues: (1) intra-feature temporal dependency, and (2) inter-feature correlation, leading to the overlook of complex coupling information in imputation. To achieve more accurate TDSI, we design a novel imputation model called TABiG, which delicately preserves the short-term, long-term, and inter-feature dependencies by attention mechanisms in a delay error-reduced bi-directional architecture. That is, it leverages GRU to model short-term temporal dependencies and adopts self-attention mechanisms hierarchically to capture long-term temporal dependencies and inter-feature correlations. The multiple self-attention mechanisms are nested in a bi-directional structure to alleviate the problem of delay errors in RNN-like structures. To facilitate model training with higher generalization, a masking strategy that mimics various extreme real missing situations beyond the simple random ones has been adopted for generating self-supervised learning tasks. Comprehensive experiments demonstrate that TABiG significantly outperforms most state-of-the-art imputation counterparts. Complementary results and source code can be accessed at https://github.com/Zhang2112105189/TABiG

1 Introduction

An enormous amount of data is being generated at all times, making time series data ubiquitous in domains such as healthcare [13], climate monitoring, financial evaluation, and so on [34, 44]. In real-world data collection environments due to faults or negligence during the data collecting, time series data may contain both sample-wise and feature-wise missing in the time intervals [27]. The missing values make the data distribution incomplete and may mislead the downstream data analysis tasks [10]. Therefore, it is of paramount importance to develop more effective imputation methods for time series data to recover the data usability and reliability.

There are two common ways of handling missing data: deletion and imputation [8]. Deleting the whole sample of features with missing values is simple, but it is harmful to the availability of data especially for the heavy missing cases. Thus, the imputation that fills the missing values based on the available observations becomes a better solution. A common goal of imputation is to assign reasonable values to the missing positions, making the overall distribution of the completed data closer to the true distribution. Mainstream TSDI approaches utilize the non-missing parts of the data and leverages the dependencies in time series and features to infer the missing parts, thereby greatly improving the data completeness and providing a solid basis for downstream tasks. Existing TSDI attempts can be roughly divided into statistical-based, traditional machine learning-based, and deep learning-based methods.

Statistical methods, such as mean imputation, median imputation, and last observation carried forward (LOCF), are relatively simple to implement, but have not taken into account the temporal and cross-feature dependencies. Traditional machine learning methods, including K-Nearest Neighbors (KNN), matrix factorization, Multiple Imputation by Chained Equations (MICE), and Autoregressive Integrated Moving Average (ARIMA), usually adopt fixed imputation strategies and thus introduce strong missing distribution heuristics to TSDI, which limit their generalization, particularly for complex data with various missing distributions. Thanks to the powerful auto-learning ability of deep models, deep imputation models have gradually become the mainstream for TSDI [23]. The deep TSDI models can be roughly categorized into three types: (1) recurrent neural network (RNN)-based methods, (2) generative model-based methods, and (3) Transformer-based methods.

RNN-based methods leverage RNN models to capture the temporal dependencies in time series data. The gated recurrent unit (GRU) based method, GRU-D [4], was proposed to estimate missing values using hidden state decay to capture past features in a smooth manner. Although it achieved outstanding performance in healthcare data, it still has several limitations when applied to other datasets. Subsequently, MRNN [33] was proposed, which uses bi-directional structures to represent distributions that cannot be captured by forward time series. As a result, it can learn imputation in a more generalized space by exposing the hidden states in both directions. BRITS [3] is also designed based on bi-directional RNNs, estimating missing values by treating them as variables and considering the correlation between features. Although the above RNN-based models capture temporal dependencies to a certain extent, they suffer from the common error propagation issue of RNNs [35, 36], which limits their efficacy in acquiring long-term temporal dependencies.

Generative models [21, 29] have shown outstanding performance in the image domain, and some efforts have been made to apply them to the task of TSDI to generate more realistic imputed values [2]. A two-stage GAN [16] imputation method was proposed by combining GAN with a new RNN unit, namely GRUI, which learns the distribution of time series data to optimize the generator's input vector.

* Corresponding Author. Email: {yqzhang, zengan}@gdut.edu.cn

Please check ArXiv or contact the authors for any appendices or supplementary material mentioned in the paper.

Subsequently, an end-to-end E2GAN [17] method was proposed to avoid the "noise" optimization stage in the previous method by using a compression and reconstruction strategy. For the case of partially labeled time-series data, a semi-supervised generating model, SSGAN [20], was proposed, which uses a semi-supervised classifier to iteratively classify unlabeled time-series data and drives the generator to estimate missing values based on observed features and data labels. However, generating models are generally difficult to train, and models based on GANs may suffer from convergence issues and mode collapse. VAE has also been widely applied in TSDI. GP-VAE [9] utilizes a Gaussian prior for time series imputation, and similar works e.g., SGP-VAE [1], and TimeVAE [6] has also been presented in the literature. But the imputation ability of VAEs is limited by the prior distribution, which may not be able to accurately capture the feature distribution of the original data and thus limits the imputation accuracy.

Transformer-based methods [15, 30] mainly focus on the use and improvement of self-attention mechanisms [22, 28]. [32] proposed an unsupervised autoencoder model named MTSIT based on Transformer, which jointly reconstructs and computes multivariate time series using unlabeled data. Regarding spatiotemporal data [19], Cross-Dimensional Self-Attention (CDSA) [18] was proposed, which is an effective imputation method that not only captures temporal dependencies but also leverages the geographic relationships among sensors to fill in missing values in time series data. To address the problem of irregularly sampled time series, a novel approach called NRTSI [24] was proposed. Furthermore, SAITS [7] utilizes two diagonal-masked multi-head attention modules for joint reconstruction and imputation. Although Transformer-based structures typically employ self-attention mechanisms to capture long-term dependencies in time series, they do not consider the temporal dependencies between adjacent elements in the sequence during the modeling process, which may limit the utilization of local structural information in the sequence [26].

Apart from the aforementioned three categories of imputation methods, self-supervised tasks are widely considered to be effective in improving the performance of deep imputation models. In most related works, artificial missing data is generated on time series to present the model with different missing rates during the training process, in order to enhance the generalization of models. However, most existing works only randomly generate scattered missing values, which result in a very uniform missing distribution. Such missing is for reflecting the true complex missing patterns of real data and thus has limited effect in self-supervised model training. Therefore, how to design self-supervised missing tasks that approximate various complex real-world missing patterns is a promising way to further enhance TSDI.

In this paper, we design a new imputation model based on bi-directional GRU (bi-GRU) architecture with multiple Multi-Head self-Attention (MHA) mechanisms, to adequately leverage temporal dependencies and capture the correlation between features. Specifically, to appropriately learn the distribution of time series data, we employ triple MHA (tri-MHA or tri-attention), where the first one acts to obtain the global information and perform preliminary imputation, which helps alleviate the problem of GRU gradient disappearance. This allows GRU to further acquire local relevant information and short-term temporal dependencies based on the preliminary imputation containing global basic information. Then, the second MHA is utilized to further capture long-term temporal dependencies and global inter-feature correlations. By simultaneously conducting the aforementioned operations to both forward and backward input data, the corresponding two directional representations are thus obtained. Subsequently, the third MHA is utilized to integrate these representations, facilitating a comprehensive information fusion. A self-supervised training strategy that generates missing in different degrees is also adopted to enhance the robustness and generalization of the proposed model. To train the model in a broader missing space, we also design a realistic masking strategy to simultaneously create conventional scattered and temporal block missing. Main contributions can be summarized into four-fold:

1. Bi-GRU architecture embedded with multiple MHAs is designed for TSDI. Such a structure considers long-term, short-term, and feature-wise dependencies, and also fuses the bi-directional data information in an attention-based proper way. It turns out that the model can better represent the temporal-wise and feature-wise coupling for more powerful TSDI.
2. Tri-MHA mechanism is employed to more thoroughly capture the potential complex dependencies in time series data. The three MHAs serve to enhance the performance of GRU, capture long-term temporal dependencies and inter-feature correlations, and comprehensively integrate the data based on the representations of bi-directional fusion, respectively. It is intuitive that such a tri-MHA design preserves more inference information for training.
3. A self-supervised training strategy with a realistic masking strategy is adopted to guide the training of the proposed imputation model. Compared with existing random scattered missing strategies, we generate missing by masking data values in a temporal block manner. It turns out that the model learns in a more thorough missing space and thus obtains a better generalization.
4. In comparison with several state-of-the-art imputation methods, the proposed Tri-Attention Bi-GRU (TABiG) model achieves significantly better imputation performance under various missing situations. Moreover, TABiG does not bring much extra computation cost during imputation, which makes it promising in supporting real applications.

2 Preliminaries

In this section, we describe some necessary preparations for TSDI. We consider a multivariate time series sample $X = \{x_1, x_2, \ldots, x_t, \ldots, x_T\} \in R^{T\times D}$ with T time steps and D feature dimensions, where $x_t = \{x_t^1, x_t^2, \ldots, x_t^d, \ldots, x_t^D\} \in R^{1\times D}$ represents the features of the d-th dimension of the t-th step in X.

Since each value in X may be missing, a mask vector $M = \{m_1, m_2, \ldots, m_t, \ldots, m_T\} \in R^{T\times D}$ is introduced to represent the positions of missing values in X, $m_t = \{m_t^1, m_t^2, \ldots, m_t^d, \ldots, m_t^D\} \in R^{1\times D}$, where

$$m_t^d = \begin{cases} 0 & \text{if } x_t^d \text{ is observed,} \\ 1 & \text{if } x_t^d \text{ is missing.} \end{cases}$$

In many time series datasets, some values may be missing for multiple consecutive time steps. To address this, for each sample X, we introduce a missing time interval matrix $\delta = \{\delta_1, \ldots, \delta_t, \ldots, \delta_T\} \in R^{T\times D}$, which represents the time interval between the current time step and the last observed time step. Where $\delta_t = \{\delta_t^1, \ldots, \delta_t^d, \ldots, \delta_t^D\} \in R^{1\times D}$, and it is calculated as follows:

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

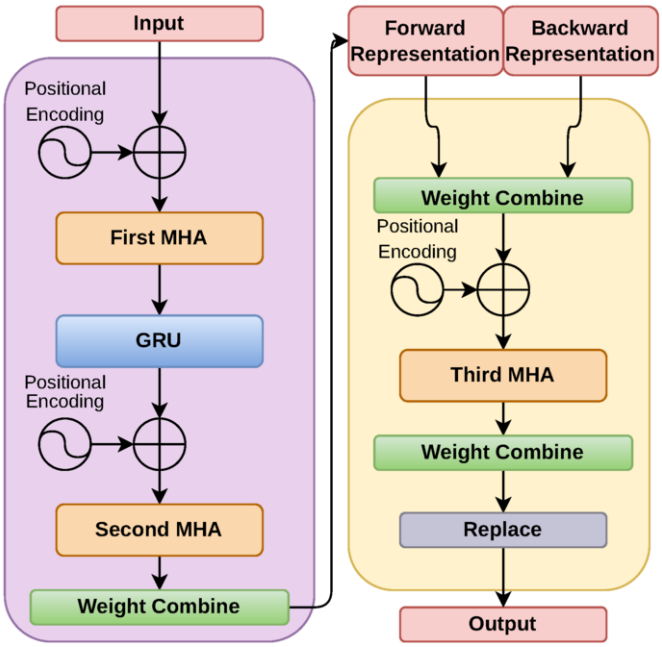

Figure 1: Forward pass of TABiG. The left frame showcases the long short-term dependency learning module. The right frame corresponds to the bi-directional information fusion module.

For the complete dataset, we apply a realistic masking strategy to create various missing percentages. We first create temporal missing by randomly selecting 10 consecutive time steps to be missing, followed by randomly masking scattered values, with each type of missing accounting for half of the missing values.

The objective of this study is to find a more accurate X' for each missing value X in an unsupervised setting. Our proposed model structure is illustrated in Figure 1. After attaching positional encoding to the input data in the long short-term dependency learning module, we perform the first MHA for initial imputation. Subsequently, the GRU is applied to capture short-term time dependencies, followed by the second MHA to further capture long-term time dependencies. The forward representations, containing both short and long-term dependencies, are obtained through a weighted combination. The process for obtaining the backward representations follows a similar procedure. Moving on to the bi-directional information fusion module, we combine the representations from both directions through weighted aggregation and utilize the third MHA for integration. Finally, another weighted combination is performed, and the missing values are replaced, resulting in reliable outcomes.

3 Methodology

Details of the proposed TABiG are described in four parts: (1) Bi-directional framework; (2) Long short-term dependency learning; (3) Bi-directional information fusion; and (4) Optimization algorithm.

3.1 Bi-directional Framework

We begin by considering a single-directional recurrent process that estimates the variables in a time series by traversing each time step. The value at the current time step is derived from the previous time step using a function, and this process is repeated for all variables. If an observation is available at the current time step, we compute the error between the estimated and observed values to verify the imputation performance. To avoid error propagation, the true observation at the current time step is passed down as input for estimating the value at the next time step. However, if the observation value is missing at the current time step, we can only pass the estimated value as the input to estimate the value at the next time step. The accumulated error between the observation value and the estimated value, called delay error, can only be obtained when the next time step with an observation value arrives.

As an example, assume a variable's time series has a length of 8, denoted as $V = \{v_1, v_2, v_3^m, v_4^m, v_5^m, v_6^m, v_7, v_8\}$, where "$m$" indicates a missing observation value. In a single-directional recurrent process, we estimate the value v_t at the current time step t using the value passed from the previous time step $t-1$. We can immediately calculate the error between the observation value and the estimated value at $t = 1$ and $t = 2$. However, when there are consecutive missing observation values at $t = 3, 4, 5, 6$, we cannot obtain the error between the observation value and the estimated value. Only when an observation value appears again at $t = 7$ can we obtain the error, which includes the errors at $t = 3, 4, 5, 6$ and the current error at $t = 7$. This error is the delay error of $v_{t=3,4,5,6}$.

The delay error caused by missing values in the recurrent process is unavoidable and can lead to the accumulation and propagation of previous errors. This can result in bias amplification, known as the bias explosion problem. Furthermore, the delay error may also be a reason for the slow convergence of RNN-based models. To alleviate such issues, we adopt a bi-directional architecture, where the values at the current time step in the time series can be derived not only from the function of the previous value but also from the function of the next value in the backward direction. That is, while the forward errors at $t = 3, 4, 5, 6$ can only be obtained at $t = 7$, the backward errors can be obtained at $t = 2$. The delay of forward errors is too long at $t = 3$, requiring four steps to obtain, while the backward errors have a much shorter delay, requiring only one step. Therefore, the bi-directional architecture can effectively mitigate the problems caused by error propagation and fully leverage temporal information from both forward and backward directions to capture the temporal dependencies of the features.

3.2 Long Short-Term Dependency Learning Module

This module aims to enhance the model's ability to capture temporal dependencies and inter-feature correlations while robustly estimating missing values. The module adopts a double MHA with GRU imputation applied between the two layers of MHA for time series data. The first MHA obtains initial global features and temporal dependencies to alleviate the issue of GRU gradient vanishing and yield informative feature representations. To comprehensively integrate the missing information, the feature vectors X, missing position matrix M, and missing time interval matrix δ are concatenated as input, which is projected into an H-dimensional space through a linear fully connected layer with positional encoding E_{pos} as

$$X_{\text{a1}} = (W_{\text{a1}}[X \circ M \circ \delta] + b_{\text{a1}}) + E_{\text{pos}} \tag{1}$$

where $W_{\text{a1}} \in R^{D \times H}$ and $b_{\text{a1}} \in R^{T \times H}$. With the incorporation of positional encoding, MHA can consider the positional relationships among elements when processing time series data, thus better capturing the sequence structure and contextual information, thereby enhancing the model's expressive capacity and generalization. X_{a1} denotes the input sequence for the first MHA. X_{a1} is mapped to query

Q, key K, and value V with dimensions d_k, d_k, and d_v, respectively:

$$Q = W_Q X_{\mathrm{a1}}, \quad K = W_K X_{\mathrm{a1}}, \quad V = W_V X_{\mathrm{a1}}, \tag{2}$$

where the corresponding parameters are $W_Q \in R^{H\times d_k}$, $W_K \in R^{H\times d_k}$, and $W_V \in R^{H\times d_v}$.

In Eq. (3), we adopt a scaled dot-product to calculate attention scores, which helps control the distribution range and enhances the model's stability, thereby promoting convergence during training. Attention scores for all heads are concatenated in Eq. (4) and projected to an H-dimensional space. By concatenating the attention scores from multiple heads, the model acquires a more diverse and enriched feature representation. Mapping the attention scores from multiple heads into an H-dimensional space allows the model to engage in feature composition and interaction within a higher-dimensional feature space. This facilitates the model's ability to capture interrelations and dependencies among input sequences, thereby improving its expressive power and learning performance. The FeedForward layer further processes and transforms the weighted feature vectors in Eq. (3.2). Leveraging non-linear transformations and increased depth, the feed-forward layer enhances the model's representational capacity and generalization, empowering it to effectively handle complex data structures and feature relationships.

$$A^i_{Q,K,V} = \mathrm{softmax}(\frac{Q^i(K^i)^\top}{\sqrt{d_k}})V = A^i_1 V \tag{3}$$

$$\mathrm{MHA}_1(X_{\mathrm{a1}}) = [A^1_{Q,K,V} \circ \ldots \circ A^i_{Q,K,V} \circ \ldots \circ A^l_{Q,K,V}]W_O \tag{4}$$

$$\mathrm{FeedForward}(\mathrm{MHA}_1(X_{\mathrm{a1}})) = W_2\,\mathrm{ReLU}(W_1\,\mathrm{MHA}_1(X_{\mathrm{a1}})+b_1)+b_2 \tag{5}$$

where $W_O \in R^{ld_v\times H}$,$W_1 \in R^{H\times d_{\mathrm{inner}}}$, $W_2 \in R^{d_{\mathrm{inner}}\times H}$, $b_1 \in R^{T\times d_{\mathrm{inner}}}$, and $b_2 \in R^{T\times H}$ are the corresponding parameters.

The entire MHA can be represented using Eq. (6), where N denotes the number of stacked layers. Eq. (7) reduces the dimensionality of the MHA representations, maps them back to the original feature dimension, and strengthens the output representation using the ReLU activation function and fully connected layer. Eq. (8) replaces the missing values in X at the missing positions with X_1.

$$X_{1\mathrm{SA}} = \{\mathrm{FeedForward}(\mathrm{MHA}_1(X_{\mathrm{a1}}))\}^N \tag{6}$$

$$X_1 = W_{\beta 1}\,\mathrm{ReLU}(W_{\alpha 1}X_{1\mathrm{SA}} + b_{\alpha 1}) + b_{\beta 1} \tag{7}$$

$$X^{\mathrm{replace}} = M \odot X + (1 - M) \odot X_1 \tag{8}$$

where $W_{\alpha 1} \in R^{H\times D}$, $W_{\beta 1} \in R^{D\times D}$, $b_{\alpha 1} \in R^{T\times D}$, $b_{\beta 1} \in R^{T\times D}$ are the corresponding parameters.

After the first MHA, we generate an initial imputation X that captures long-term temporal dependencies and inter-feature correlations. Then, we use GRU for further imputation. Firstly, we iterate through the time steps of X^{replace}, missing position matrix M, and missing time interval matrix δ to obtain x^{replace}_t, m_t and δ_t. In GRU, the hidden state is continually updated and passed to the next time step, but its ability to capture long-term dependencies is limited as the information contained in the hidden state will gradually be diluted and forgotten. To address this issue and improve the model's robustness and generalization ability, we introduce a time decay factor based on the missing time interval in Eq. (9) and dynamically adjust the hidden state decay in Eq. (10) to better retain information from hidden states with smaller missing time intervals and decay information from hidden states with larger missing time intervals more quickly.

$$\eta_t = \exp\{-\max(0, W_\eta \delta_t + b_\eta)\} \tag{9}$$

$$h_{t-1} = h_{t-1} \odot \eta_t \tag{10}$$

where $W_\eta \in R^{D\times H}$ and $b_\eta \in R^{1\times H}$ are corresponding parameters.

Eq. (11) maps the decayed hidden state back to the original feature dimension. In Eq. (12), we use a parameter matrix with diagonal zeros to estimate each feature based on the other features.

$$x^{\mathrm{history}}_t = W_{\mathrm{history}} h_{t-1} + b_{\mathrm{history}} \tag{11}$$

$$x^{\mathrm{feature}}_t = W_{\mathrm{feature}} x^{\mathrm{replace}}_t + b_{\mathrm{feature}} \tag{12}$$

where $W_{\mathrm{history}} \in R^{H\times D}$, $b_{\mathrm{history}} \in R^{1\times D}$,$W_{\mathrm{feature}} \in R^{D\times D}$, and $b_{\mathrm{feature}} \in R^{1\times D}$ are the corresponding parameters.

Eq. (13) calculates the average weight of the weights from the first MHA. In Eq. (14), we learn a weight based on the time decay factor, missing position matrix, and the average weight of the MHA, and adaptively combine the feature-based estimation and the hidden state estimation in Eq. (15) to obtain the imputation result for the current time step of GRU. To prevent error propagation, we only replace the missing values with the imputation result at the missing positions to obtain c_{replace} in Eq. (16), which is used as the input of the GRU.

$$\hat{A}_1 = \frac{1}{k}\sum_{i=1}^{k} A^i_1 \tag{13}$$

$$\lambda_{\mathrm{hf}} = \mathrm{sigmoid}(W_{\mathrm{hf}}[\eta_t \circ m_t \circ \hat{A}_1] + b_{\mathrm{hf}}) \tag{14}$$

$$c_t = \lambda_{\mathrm{hf}} \odot x^{\mathrm{feature}}_t + (1 - \lambda_{\mathrm{hf}}) \odot x^{\mathrm{history}}_t \tag{15}$$

$$c_{\mathrm{replace}} = m_t \odot x_t + (1 - m_t) \odot c_t \tag{16}$$

where $W_{\mathrm{hf}} \in R^{(2D+T)\times D}$ and $b_{\mathrm{hf}} \in R^{1\times D}$ are the corresponding parameters. W_{feature} is a parameter matrix with diagonal zeros.

Within the GRU, we first compute the update gate, as shown in Eq. (17), to regulate the contribution of the previous hidden state h_{t-1} to the current hidden state. The reset gate, as calculated in Eq. (18), is then used to control the contribution of the past hidden state h_{t-1} to the candidate hidden state, taking into account the current input c_{replace}. Based on the reset gate, current input c_{replace}, and past hidden state h_{t-1}, we obtain the candidate hidden state using Eq. (19). Finally, in Eq. (20), we calculate the hidden state h_t, which will be passed to the next time step, by combining the update gate and candidate hidden state.

$$z_t = \sigma(W_z c_{\mathrm{replace}} + U_z h_{t-1} + b_z) \tag{17}$$

$$r_t = \sigma(W_r c_{\mathrm{replace}} + U_r h_{t-1} + b_r) \tag{18}$$

$$\tilde{h}_t = \tanh(W_h c_{\mathrm{replace}} + U_h(r_t \odot h_{t-1}) + b_h) \tag{19}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \tag{20}$$

where $W_z \in R^{D\times H}$, $U_z \in R^{H\times H}$, $b_z \in R^{1\times H}$, $W_r \in R^{D\times H}$, $U_r \in R^{H\times H}$, $b_r \in R^{1\times H}$, $W_h \in R^{D\times H}$, $U_h \in R^{H\times H}$, and $b_h \in R^{1\times H}$ are the corresponding parameters.

We concatenate the outputs c_t ($t \in \{1, 2, \ldots, T\}$) of each time step of the GRU in Eq. (21) and use a FeedForward layer to enhance the expression ability. In Eq. (22), we replace the missing values with X_{gru} to obtain R_{gru}, which serves as the input for the second MHA. After obtaining short-term dependencies via GRU, we leverage the second MHA to further capture long-term temporal dependencies and correlations between features based on the short-term dependencies. Eq. (23) to (25) describe the process of the second MHA, which is similar to the first one and requires no further elaboration.

$$X_{\mathrm{gru}} = \mathrm{FeedForward}([c_1 \circ c_2 \circ \ldots \circ c_t \circ \ldots \circ c_T]) \tag{21}$$

$$R_{\text{gru}} = M \odot X + (1 - M) \odot X_{\text{gru}} \tag{22}$$

$$X_{\text{a2}} = (W_{\text{a2}}[R_{\text{gru}} \circ M \circ D] + b_{\text{a2}}) + E_{\text{pos}} \tag{23}$$

$$X_{\text{2SA}} = \{\text{FeedForward}(\text{MHA}_2(X_{\text{a2}}))\}^N \tag{24}$$

$$X_2 = W_{\beta 2}\,\text{ReLu}(W_{\alpha 2} X_{\text{2SA}} + b_{\alpha 2}) + b_{\beta 2} \tag{25}$$

where $W_{\text{a2}} \in R^{D\times H}$, $b_{\text{a2}} \in R^{T\times H}$, $W_{\alpha 2} \in R^{H\times D}$, $W_{\beta 2} \in R^{D\times D}$, $b_{\alpha 2} \in R^{T\times D}$, and $b_{\beta 2} \in R^{T\times D}$.

Eq. (26) calculates the average weight of the second MHA, and in Eq. (27), we concatenate it with the missing position information and learn a combination weight λ_{ls}. To effectively leverage the strengths of both GRU and MHA, and alleviate the issue of information loss to enhance model robustness and imputation performance, we use a combination weight λ_{ls} in Eq. (28) to weight and combine the two representations, obtaining the final forward representation X_{forward}.

$$\hat{A}_2 = \frac{1}{k}\sum_{i=1}^{k} A_2^i \tag{26}$$

$$\lambda_{\text{ls}} = \text{sigmoid}(W_{\text{ls}}[m_t \circ \hat{A}_2] + b_{\text{ls}}) \tag{27}$$

$$X_{\text{forward}} = \lambda_{\text{ls}} \odot C_x + (1 - \lambda_{\text{ls}}) \odot X_2 \tag{28}$$

where $W_{\text{ls}} \in R^{(D+T)\times D}$ and $b_{\text{ls}} \in R^{T\times D}$ are the corresponding parameters. The forward representation is obtained for forward input, while the backward representation is obtained through the same process for backward input. The single-directional representation not only captures long- and short-term temporal dependencies in the respective direction but also the inter-feature correlations.

3.3 Bi-directional Information Fusion Module

In the previous module, we obtained the representation that captures the long-term and short-term dependencies and feature correlations in a single direction. However, many works that utilize bi-directional structures simply add and average the bi-directional representations, which can result in information loss and bias, ultimately affecting the imputation performance. To avoid these issues and more effectively combine the bi-directional information to obtain a more accurate and reliable imputation result, we use adaptive and learnable weights.

From the bi-directional imputation processes, we obtain the forward representations X_{forward}, the backward representations X_{backward}, and the average weights of the second MHA. The MHA weights obtained by the long short-term dependency learning module contain rich comprehensive information, reflecting the dependency relationships and feature correlations between different time steps and the relative importance between them, providing contextual information in the sequence. Therefore, we learn the weights based on the bi-directional attention weights and missing information in Eq. (29) and use the learned weights to combine the bi-directional representations in Eq. (30). In Eq. (31), we replace the missing values with $X_{\text{bidirection}}$ to obtain $R_{\text{bidirection}}$ as the input for the third MHA.

$$\lambda_{\text{fb}} = \text{sigmoid}(W_{\text{fb}}[m_t \circ \hat{A}_{\text{forward}} \circ \hat{A}_{\text{backward}}] + b_{\text{fb}}) \tag{29}$$

$$X_{\text{bidirection}} = \lambda_{\text{fb}} \odot X_{\text{forward}} + (1 - \lambda_{\text{fb}}) \odot X_{\text{backward}} \tag{30}$$

$$R_{\text{bidirection}} = M \odot X + (1 - M) \odot X_{\text{bidirection}} \tag{31}$$

where $W_{\text{fb}} \in R^{(D+2T)\times D}$ and $b_{fb} \in R^{T\times D}$.

The bi-directional combination representations encompass a more comprehensive representation of the sequential information, incorporating both past historical context and future predictive context. To effectively capture the sequential structure and contextual information, and enhance the model's representational capacity and imputation performance, we employ the third MHA to integrate these information sources. This integration step enables the model to better exploit the temporal dependencies and inter-feature dependencies within the sequence, thereby enhancing its expressive power and imputation effectiveness. Eq. (32)-(34) describes the process of the third MHA, which is similar to the previous two.

$$X_{\text{a3}} = (W_{\text{a3}}[R_{\text{bidirection}} \circ M] + b_{\text{a3}}) + E_{\text{pos}} \tag{32}$$

$$X_{\text{3SA}} = \{\text{FeedForward}(\text{MHA}_3(X_{\text{a3}}))\}^N \tag{33}$$

$$X_3 = W_{\beta 3}\,\text{ReLu}(W_{\alpha 3} X_{\text{3SA}} + b_{\alpha 3}) + b_{\beta 3} \tag{34}$$

where $W_{\text{a3}} \in R^{D\times H}$, $b_{\text{a3}} \in R^{T\times H}$, $W_{\alpha 3} \in R^{H\times D}$, $W_{\beta 3} \in R^{D\times D}$, $b_{\alpha 3} \in R^{T\times D}$, and $b_{\beta 3} \in R^{T\times D}$.

To mitigate the loss of useful information when integrating information using MHA, we combine it with the representation of bi-directional fusion using weighted averaging as follows

$$\hat{A}_3 = \frac{1}{k}\sum_{i=1}^{k} A_3^i \tag{35}$$

$$\lambda_{\text{combine}} = \text{sigmoid}(W_{\text{combine}}[m_t \circ \widehat{A}_3] + b_{\text{combine}}) \tag{36}$$

$$X_{\text{final}} = \lambda_{\text{combine}} \odot X_3 + (1 - \lambda_{\text{combine}}) \odot X_{\text{bidirection}} \tag{37}$$

$$X_{\text{out}} = M \odot X + (1 - M) \odot X_{\text{final}} \tag{38}$$

where $W_{\text{combine}} \in R^{(D+T)\times D}$ and $b_{\text{combine}} \in R^{T\times D}$ are the corresponding parameters. Eq. (35) calculates the average weight of the MHA for the third layer, and Eq. (36) concatenates the average weight with the missing position matrix to learn a combination weight λ_{combine}. Then, Eq. (37) combines the representations of the third layer of multi-head self-attention with the bi-directional fusion representation based on the learned weight. Finally, Eq. (38) replaces the missing values to obtain the final imputation result.

3.4 Optimization Algorithm

We adopt Mean Absolute Error (MAE) to form our loss function as MAE is robust in handling outliers and noise. Specifically, MAE does not square the differences, thus preventing the enlargement of the impact of extreme values. Moreover, MAE directly measures the average absolute deviation between the predicted and true values, enhancing interpretability and providing a more intuitive assessment of the model's performance when evaluating its effectiveness. We use $l_{\text{MAE}}(X_i, X, M)$ to represent the mean absolute error between the input sequence X_i and the ground truth sequence X on the masked set M, and the loss function consists of three components: the self-supervised masked imputation loss, the reconstruction loss, and the bi-directional consistency loss. The self-supervised masked imputation loss in Eq. (39) drives the model to better understand the inter-feature relationships and temporal dependencies of features. As a result, the loss encourages the model to learn the time series' underlying structure, thus leading to more robust representations, stabilizing the training process, and improving imputation performance. $M_{\text{indicating}}$ is the matrix that identifies the artificial missing position. The bi-directional consistency loss in Eq. (40) penalizes inconsistencies or discrepancies in two directions to make the model generate coherent and consistent estimations throughout the entire time series, which contributes to preserving the temporal dependencies and enhancing the imputation accuracy.

$$L_{\text{mask}} = l_{\text{MAE}}(X_{\text{final}}, X, M_{\text{indicating}}) \tag{39}$$

$$L_{\text{bidirection}} = l_{\text{MAE}}(X_{\text{forward}}, X_{\text{backward}}) \tag{40}$$

To expedite the model convergence, we calculate the reconstruction loss for multiple modules and aggregate the reconstruction loss by computing their average. The reconstruction loss comprises four components: the forward reconstruction loss, backward reconstruction loss, bi-directional combination reconstruction loss, and final result reconstruction loss. Eq. (41) represents the forward reconstruction loss, which includes the loss of the first layer of multi-head self-attention, the GRU process loss in Eq. (42), and the loss of the second layer of multi-head self-attention.

$$L_{\text{forward}} = \frac{1}{3}\left(l_{\text{MAE}}(X_1, X, M) + L_{\text{GRU}} + l_{\text{MAE}}(X_2, X, M)\right) \tag{41}$$

$$\begin{aligned} L_{\text{GRU}} = \frac{1}{3T}\sum_{t=1}^{T}(l_{\text{MAE}}(x_t^{\text{history}}, x_t, m_t) + \\ l_{\text{MAE}}(x_t^{\text{feature}}, x_t, m_t) + l_{\text{MAE}}(c_t, x_t, m_t)) \end{aligned} \tag{42}$$

The backward reconstruction loss is similar to the forward reconstruction loss. The total reconstruction loss in Eq. (43) combines the reconstruction losses corresponding to different modules, which allows benefiting from compensation knowledge provided by each module. It turns out that the effects of outliers and extreme errors introduced by individual modules can be weakened to improve the robustness of imputation. The overall loss is shown in Eq. (44).

$$\begin{aligned} L_{\text{reconstruction}} = \frac{1}{4}(L_{\text{forward}} + L_{\text{backward}} + \\ l_{\text{MAE}}(X_{\text{bidirection}}X, M) + l_{\text{MAE}}(X_{\text{final}}, X, M)) \end{aligned} \tag{43}$$

$$L_{\text{Totally}} = L_{\text{mask}} + L_{\text{bidirection}} + L_{\text{reconstruction}} \tag{44}$$

4 Experiments

We conduct experiments using three evaluation metrics on four benchmark datasets to compare the proposed TABiG against eight counterparts. Specifically, we performed four experiments: (1) Imputation performance comparison, which compares our TABiG with state-of-the-art methods on four datasets to verify the effectiveness; (2) Ablation experiments to analyze the effectiveness of the proposed modules in the TABiG architecture; (3) Efficiency evaluation by comparing the execution time of TABiG and other state-of-the-art methods; (4) Performance under different missing rates (the corresponding results are included in the supplementary material due to space limitation). We first introduce datasets, counterparts, and experimental settings, and then demonstrate the experimental results with observations.

4.1 Datasets

General information of the datasets is shown in Table 1. To evaluate the imputation performance, we randomly mask 10% of the data values for both the test and validation sets across all datasets, to form the ground truth. Then we describe the usage of different datasets according to their source papers below.

- PhysioNet 2012 [11] comprises records from 12,000 ICU patients admitted and monitored for 48 hours following admission, measuring 35 time-series variables, such as Respiration rate and Heart rate. Due to the irregularity of the sampling schedule, the dataset is highly sparse, with 80% missing values. We adopt a five-fold cross-validation approach and report average performance, by randomly selecting 80% of the samples as training set and 20% as test set, and further partitioning 20% of the training set as a validation set for every single implementation.
- BeiJing PM2.5 [31,43] includes hourly PM2.5 concentration measurements from 36 monitoring stations in Beijing. It covers a 12-month period from May 1, 2014, to April 30, 2015, with 36 features and 13.2% missing values. For testing, we use the 3rd, 6th, 9th, and 12th months, while the 4th and 7th months are used as validation sets, and the remaining data as training sets. Each time series sample is a continuous sequence of 36-time steps. We repeat experiments 5 times and report the average performance.
- Air Quality [37] consists of hourly air pollutant data collected from 12 monitor stations in Beijing over 48 months from March 1, 2013, to February 28, 2017. Each station measures 11 time-varying variables e.g., PM10 and SO2. By concatenating the time series variables measured by the 12 stations, the dataset contains 132 features with a missing rate 1.6%. The first 10, middle 10, and rest months of the data are used as test set, validation set, and training set, respectively. We selected a continuous 24-time step as a time series sample. We repeat the experiments five times and report the average performance.
- Localization [14] contains activity records of five persons performing walking, falling, sitting, etc. (11 activities in total). Each person wears several sensors, with each sensor recording a 3-dimensional coordinate every 20 to 40 milliseconds. We encode the sensors and their 3-dimensional coordinates as features, resulting in four features. As this dataset is complete, we mask the data at different percentages (10%, 30%, 50%, 70%, 90%) by realistic masking to generate value missing. We select 40 consecutive time steps as a time series sample, and adopt the same five-fold cross-validation setting as that of PhysioNet 2012.

Properties	PhysioNet 2012	BeiJing PM2.5	Air Quality	Localization
No. samples	11987	242	1461	4110
No. features	38	36	132	4
Sequence length	48	36	24	40
Missing rate	80%	13.2%	1.6%	0%

Table 1: General information of the four datasets.

Hyper-parameters	PhysioNet 2012	BeiJing PM2.5	Air Quality	Localization
Batch Size	128	32	128	128
Hidden Layer Size	256	128	512	256
No. Attention Group	5	2	1	2
No. Attention head	8	4	4	4

Table 2: Experimental settings about hyper-parameters.

4.2 Counterparts

We compared a total of nine methods, including three conventional methods, five advanced deep learning methods, and one proposed method. To ensure a fair comparison, we maintained the hyper-parameters recommended in the source papers if they use the same dataset as ours; otherwise, we keep them consistent with the hyper-parameters of our model.

The eight compared methods are briefly introduced below: (1) Mean imputation: replacing missing values with the mean of the corresponding samples; (2) Median imputation: replacing missing values with the median of the corresponding samples; (3) KNN imputation: for each data sample, we find its k nearest neighbors using Euclidean distance and estimate missing values using the

Methods	PhysioNet 2012	BeiJing PM2.5	Air Quality	Localization	Ave. Rank
Mean	0.4624±0.003	0.5217±0.003	0.3654±0.001	0.5990±0.007	8.25
Median	0.4493±0.003	0.5108±0.004	0.3585±0.001	0.5777±0.007	7.25
KNN	0.5142±0.002	0.4756±0.002	0.3259±0.002	0.5408±0.008	7.00
MRNN	0.5551±0.007	0.3295±0.005	0.2914±0.002	0.8104±0.007	7.50
BRITS	0.2654±0.028	0.1813±0.003	0.1443±0.001	0.3769±0.006	3.50
Transformer	0.2079±0.002	0.2100±0.005	0.1567±0.001	0.1790±0.007	3.25
MTSIT	0.3814±0.005	0.2476±0.006	0.2119±0.002	0.2608±0.004	4.75
SAITS	0.2061±0.002	0.1928±0.003	0.1409±0.003	0.1814±0.009	2.50
TABiG (ours)	**0.2021±0.002**	**0.1783±0.003**	**0.1227±0.001**	**0.1628±0.008**	**1.00**

Table 3: Comparison of imputation performance on MAE. The best-performing method is highlighted in **bold**.

Methods	PhysioNet 2012	BeiJing PM2.5	Air Quality	Localization	Ave. Rank
Mean	0.6501±0.002	0.5822±0.005	0.5159±0.002	0.7243±0.008	8.25
Median	0.6317±0.002	0.5701±0.007	0.5061±0.002	0.6986±0.008	7.25
KNN	0.7229±0.003	0.5307±0.002	0.4601±0.003	0.6539±0.007	7.00
MRNN	0.7804±0.006	0.3678±0.005	0.4114±0.003	0.9800±0.003	7.50
BRITS	0.3730±0.037	0.2023±0.003	0.2038±0.001	0.4558±0.004	3.50
Transformer	0.2923±0.003	0.2344±0.005	0.2213±0.002	0.2164±0.008	3.25
MTSIT	0.5363±0.009	0.2764±0.007	0.2992±0.003	0.3154±0.003	4.75
SAITS	0.2898±0.002	0.2152±0.003	0.1989±0.004	0.2193±0.010	2.50
TABiG (ours)	**0.2842±0.003**	**0.1990±0.003**	**0.1733±0.002**	**0.1968±0.009**	**1.00**

Table 4: Comparison of imputation performance on MRE. The best-performing method is highlighted in **bold**.

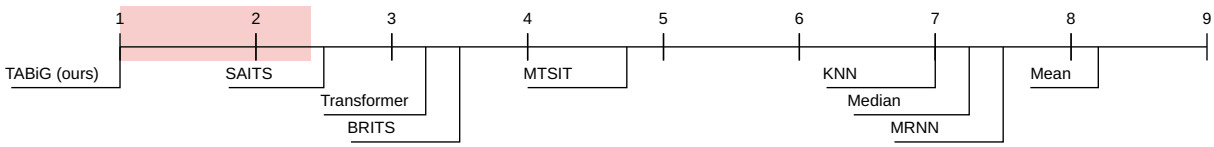

Figure 2: Significance evaluation using Bonferroni-Dunn test at confidence interval 95% (i.e. $\alpha = 0.05$). The pink region stands for the right side of the critical difference interval. It is worth noting that counterparts ranked outside the pink region are considered to have significantly different performance in comparison with the proposed TABiG.

weighted average of its neighbors; (4) MRNN [33]: using a multi-directional recurrent neural network to insert missing values and estimate them across the data stream; (5) BRITS [3]: this method uses a bi-directional LSTM with history regression and feature regression to estimate missing values; (6) Transformer: using the Transformer's encoder for missing value estimation; (7) MTSIT [32]: using learnable position encoding and the Transformer's encoder for missing value estimation; (8) SAITS [7]: using two diagonal-masked multi-head attention modules for joint reconstruction.

4.3 Experimental Setup

In the proposed model, we set the learning rate to 0.001 and hyper-parameters w.r.t. datasets are shown in Table 2. Early stopping is applied to all models, and training is stopped if the mean absolute error (MAE) does not decrease for 30 epochs. We train our model using the Adam optimizer on an Nvidia GeForce RTX 3090 GPU, and the implementation is based on PyTorch.

For the experimental results, we conduct five-fold cross-validation on PhysioNet 2012 and Localization, and five repetitions on BeiJing PM2.5 and Air Quality. We report the average results on the five runs w.r.t. each dataset. We use three metrics, i.e., MAE, MRE, and RMSE, to evaluate the imputation performance. Results w.r.t. RMSE is reported as complementary experimental results, which can be found through the link provided at the end of the Abstract.

4.4 Imputation Performance

Table 3 and Table 4 show the imputation performance of all compared methods on four real datasets w.r.t. MAE and MRE.

The experimental results demonstrate that traditional imputation methods, including mean imputation, median imputation, and KNN imputation, perform relatively worse, indicating that they are incompetent in handling the imputation of complex time-series data. For datasets such as Beijing PM2.5 and Air Quality with low missing rates and gradual temporal distribution changing, RNN-based methods like MRNN and BRITS exhibit satisfactory performance by leveraging the intrinsic advantages of RNN. However, their performance significantly deteriorates on datasets with high missing rates like PhysioNet 2012, and datasets with sudden distribution changes like Localization, a dataset recording human activities. Since BRITS more comprehensively considers feature correlations, it performs well among the RNN-based approaches.

Benefiting from the global information exploitation capability provided by the self-attention mechanism, Transformer, MTSIT, and SAITS demonstrate advantages in handling the more challenging PhysioNet 2012 and Localization datasets. The proposed TABiG, a fusion of self-attention and GRU, demonstrates superior imputation performance on all the datasets. We conducted a significance evaluation on all the compared methods by adopting the Bonferroni-Dunn test with critical interval as described in [5]. The corresponding results are demonstrated in Figure 2, which illustrates that the proposed TABiG significantly outperforms all the counterparts (excluding SAITS). In the next subsection, we further conduct an ablation study to illustrate the effectiveness of different modules of TABiG.

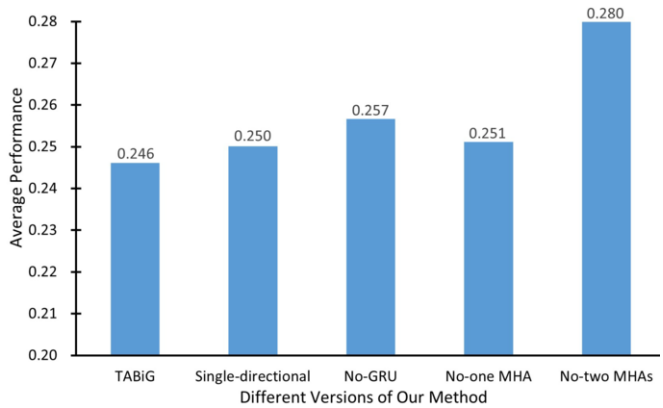

Figure 3: Comparison of TABiG with its different versions, i.e., TABiG with a single direction, without GRU, without the first MHA, and without the former two MHAs.

Methods	Test Time (s)	Sample Time (ms)
MRNN	1.711	0.714
BRITS	4.571	1.906
Transformer	1.024	0.427
MTSIT	2.482	1.035
SAITS	1.219	0.508
TABiG (ours)	3.742	1.561

Table 5: Comparison of execution time. Test Time refers to the overall execution time taken to perform imputation on the dataset, and Sample Time represents the average processing time per sample.

4.5 Ablation Experiment

We conducted ablation experiments by removing different components of TABiG to form different ablated versions, and the comparative results on these versions are shown in Figure 3. Specifically, we removed the bi-directional structure, GRU layer, the first MHA, and the former two MHAs. For the convenience of observation, the performance of all the compared ablated versions on the four datasets in terms of three validity indices is averaged and reported. It can be observed that the performance of the complete version of our method TABiG outperforms all the ablated versions, which illustrates the effectiveness of combining all the proposed modules. Performance of TABiG with a single direction and TABiG without GRU getting worse, illustrating the reasonableness of adopting the bi-GRU structure. Moreover, TABiG without the first MHA performs better than TABiG without the former two MHAs, which verifies the soundness of our arrangement of the MHAs.

4.6 Model Execution Time

To evaluate the efficiency of the proposed TABiG, the execution time of all the compared state-of-the-art methods are compared on PhysioNet 2012 with 2398 time-series samples. The overall execution time and the average time per example are reported in Table 5. As our method involves a more elaborately designed architecture, it exhibits a relatively higher computation cost compared to the methods solely based on self-attention. Nonetheless, it demonstrates better efficiency compared to BRITS, which relies on a bi-directional RNN architecture. In general, TABiG does not bring much extra computation cost in comparison to the existing state-of-the-art methods, which illustrates its potential in real time-series imputation applications.

5 Concluding Remarks

A new imputation model TABiG that leverages tri-MHA and bi-GRU to impute missing values in multivariate time series data has been proposed. TABiG integrates information from multiple perspectives to consider long-term and short-term time dependencies in both forward and backward temporal directions, and also the correlation between features. The joint optimization across multiple modules improves the reliability of imputed values, and we also present a self-supervised learning strategy, which introduces generated realistic missing at different degrees to enhance the robustness and generalization of TABiG. Experimental results show its superiority over the state-of-the-art imputation methods.

This work mainly focuses on the imputation of numerical time-series data, which can subsequently support many significant unsupervised tasks, including clustering [39], ranking [45], etc. From the perspective of data science, more complex factors including heterogeneous features [42] [40], concept drifts [12], distribution imbalance [25], graph relationship [41] [38], etc., can be jointly considered with the data missing issue in our future work. Moreover, improving the efficiency and studying the parameter settings of TABiG would also be promising for improving its scalability in real applications.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62102097, the Key-Area R&D Program of Guangdong Province under Grant 2021B0101220006, the Natural Science Foundation of Guangdong Province under Grants: 2023A1515012855, 2023A1515012884, 2022A1515011592, and 2021A1515012300, and the Science and Technology Program of Guangzhou under grant 202201010548.

References

[1] Matthew Ashman, Jonathan So, Will Tebbutt, Vincent Fortuin, Michael Pearce, and Richard E Turner, 'Sparse gaussian process variational autoencoders', *arXiv preprint arXiv:2010.10177*, (2020).

[2] Eoin Brophy, Zhengwei Wang, Qi She, and Tomas Ward, 'Generative adversarial networks in time series: A survey and taxonomy', *arXiv preprint arXiv:2107.11098*, (2021).

[3] Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li, 'Brits: Bidirectional recurrent imputation for time series', in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 1–11, (2018).

[4] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu, 'Recurrent neural networks for multivariate time series with missing values', *Scientific Reports*, **8**(1), 6085, (2018).

[5] Janez Demšar, 'Statistical comparisons of classifiers over multiple data sets', *The Journal of Machine Learning Research*, **7**, 1–30, (2006).

[6] Abhyuday Desai, Cynthia Freeman, Zuhui Wang, and Ian Beaver, 'Timevae: A variational auto-encoder for multivariate time series generation', *arXiv preprint arXiv:2111.08095*, (2021).

[7] Wenjie Du, David Côté, and Yan Liu, 'Saits: Self-attention-based imputation for time series', *Expert Systems with Applications*, **219**, 119619, (2023).

[8] Chenguang Fang and Chen Wang, 'Time series data imputation: A survey on deep learning approaches', *arXiv preprint arXiv:2011.11347*, (2020).

[9] Vincent Fortuin, Dmitry Baranchuk, Gunnar Rätsch, and Stephan Mandt, 'Gp-vae: Deep probabilistic time series imputation', in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pp. 1651–1661, (2020).

[10] Ge Gao, Qitong Gao, Xi Yang, Miroslav Pajic, and Min Chi, 'A reinforcement learning-informed pattern mining framework for multivariate time series classification', in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 2994–3000, (2022).

[11] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley, 'Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals', *Circulation*, **101**(23), e215–e220, (2000).

[12] Ömer Gözüaçık, Alican Büyükçakır, Hamed Bonab, and Fazli Can, 'Unsupervised concept drift detection with a discriminative classifier', in *Proceedings of the International Conference on Information and Knowledge Management*, pp. 2365–2368, (2019).

[13] Junfeng Hu, Zhencheng Fan, Jun Liao, and Li Liu, 'Predicting long-term skeletal motions by a spatio-temporal hierarchical recurrent network', in *Proceedings of the European Conference on Artificial Intelligence*, 2720–2727, (2020).

[14] Boštjan Kaluža, Violeta Mirchevska, Erik Dovgan, Mitja Luštrek, and Matjaž Gams, 'An agent-based approach to care in independent living', in *Proceedings of the International Joint Conference on Ambient Ingelligence*, pp. 177–186, (2010).

[15] Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long, 'Non-stationary transformers: Exploring the stationarity in time series forecasting', in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 1–13, (2022).

[16] Yonghong Luo, Xiangrui Cai, Ying Zhang, Jun Xu, et al., 'Multivariate time series imputation with generative adversarial networks', in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 1–11, (2018).

[17] Yonghong Luo, Ying Zhang, Xiangrui Cai, and Xiaojie Yuan, 'E2gan: End-to-end generative adversarial network for multivariate time series imputation', in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 3094–3100, (2019).

[18] Jiawei Ma, Zheng Shou, and Alireza et al. Zareian, 'Cdsa: cross-dimensional self-attention for multivariate, geo-tagged time series imputation', *arXiv preprint arXiv:1905.09904*, (2019).

[19] Chuizheng Meng, Hao Niu, Guillaume Habault, Roberto Legaspi, Shinya Wada, Chihiro Ono, and Yan Liu, 'Physics-informed long-sequence forecasting from multi-resolution spatiotemporal data', in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 2189–2195, (2022).

[20] Xiaoye Miao, Yangyang Wu, and Jun et al. Wang, 'Generative semi-supervised learning for multivariate time series imputation', in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 8983–8991, (2021).

[21] Qingjian Ni and Xuehan Cao, 'Mbgan: An improved generative adversarial network with multi-head self-attention and bidirectional rnn for time series imputation', *Engineering Applications of Artificial Intelligence*, **115**, 105232, (2022).

[22] Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison Cottrell, 'A dual-stage attention-based recurrent neural network for time series prediction', *arXiv preprint arXiv:1704.02971*, (2017).

[23] Muhammad Saad, Mohita Chaudhary, Lobna Nassar, Fakhri Karray, and Vincent Gaudet, 'Versatile deep learning based application for time series imputation', in *Proceedings of the International Joint Conference on Neural Networks*, pp. 1–8, (2021).

[24] Siyuan Shan, Yang Li, and Junier B Oliva, 'Nrtsi: Non-recurrent time series imputation', *arXiv preprint arXiv:2102.03340*, (2021).

[25] Naman D Singh and Abhinav Dhall, 'Clustering and learning from imbalanced data', *arXiv preprint arXiv:1811.00972*, (2018).

[26] Qiuling Suo, Weida Zhong, Guangxu Xun, Jianhui Sun, Changyou Chen, and Aidong Zhang, 'Glima: Global and local time series imputation with multi-directional attention learning', in *Proceedings of the International Conference on Big Data*, pp. 798–807, (2020).

[27] Xianfeng Tang, Huaxiu Yao, Yiwei Sun, Charu Aggarwal, Prasenjit Mitra, and Suhang Wang, 'Joint modeling of local and global temporal dynamics for multivariate time series forecasting with missing values', in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 5956–5963, (2020).

[28] Ashish Vaswani, Noam Shazeer, and Niki et al. Parmar, 'Attention is all you need', in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 1–11, (2017).

[29] Senzhang Wang, Jiyue Li, Hao Miao, Junbo Zhang, Junxing Zhu, and Jianxin Wang, 'Generative-free urban flow imputation', in *Proceedings of the International Conference on Information & Knowledge Management*, pp. 2028–2037, (2022).

[30] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long, 'Timesnet: Temporal 2d-variation modeling for general time series analysis', *arXiv preprint arXiv:2210.02186*, (2022).

[31] Xiuwen Yi, Yu Zheng, Junbo Zhang, and Tianrui Li, 'St-mvl: filling missing values in geo-sensory time series data', in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 2704–2710, (2016).

[32] A Yarkın Yıldız, Emirhan Koç, and Aykut Koç, 'Multivariate time series imputation with transformers', *IEEE Signal Processing Letters*, **29**, 2517–2521, (2022).

[33] Jinsung Yoon, William R Zame, and Mihaela van der Schaar, 'Estimating missing data in temporal data streams using multi-directional recurrent neural networks', *IEEE Transactions on Biomedical Engineering*, **66**(5), 1477–1490, (2018).

[34] Hongyuan Yu, Ting Li, Weichen Yu, Jianguo Li, Yan Huang, Liang Wang, and Alex Liu, 'Regularized graph structure learning with semantic knowledge for multi-variates time-series forecasting', *arXiv preprint arXiv:2210.06126*, (2022).

[35] Cheng Zhang, Qiuchi Li, Lingyu Hua, and Dawei Song, 'Assessing the memory ability of recurrent neural networks', in *Proceedings of the European Conference on Artificial Intelligence*, 1658–1665, (2020).

[36] Shao-Qun Zhang and Zhi-Hua Zhou, 'Harmonic recurrent process for time series forecasting', in *Proceedings of the European Conference on Artificial Intelligence*, 1714–1721, (2020).

[37] Shuyi Zhang, Bin Guo, Anlan Dong, Jing He, Ziping Xu, and Song Xi Chen, 'Cautionary tales on air-quality improvement in beijing', *Royal Society A: Mathematical, Physical and Engineering Sciences*, **473**(2205), 20170457, (2017).

[38] Yiqun Zhang and Yiu-ming Cheung, 'A fast hierarchical clustering approach based on partition and merging scheme', in *Proceedings of the International Conference on Advanced Computational Intelligence*, pp. 846–851, (2018).

[39] Yiqun Zhang and Yiu-ming Cheung, 'Learnable weighting of intra-attribute distances for categorical data clustering with nominal and ordinal attributes', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **44**(7), 3560–3576, (2021).

[40] Yiqun Zhang and Yiu-Ming Cheung, 'Graph-based dissimilarity measurement for cluster analysis of any-type-attributed data', *IEEE Transactions on Neural Networks and Learning Systems*, (2022).

[41] Yiqun Zhang, Yiu-ming Cheung, and Yang Liu, 'Quality preserved data summarization for fast hierarchical clustering', in *Proceedings of the International Joint Conference on Neural Networks*, pp. 4139–4146, (2016).

[42] Yiqun Zhang, Yiu-ming Cheung, and An Zeng, 'Het2hom: representation of heterogeneous attributes into homogeneous concept spaces for categorical-and-numerical-attribute data clustering', in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 3758–3765, (2022).

[43] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang, 'Urban computing: concepts, methodologies, and applications', *ACM Transactions on Intelligent Systems and Technology*, **5**(3), 1–55, (2014).

[44] Mingjie Zhou, Michael Ng, Zixin Cai, and Ka Chun Cheung, 'Self-attention-based fully-inception networks for continuous sign language recognition', in *Proceedings of the European Conference on Artificial Intelligence*, 2832–2839, (2020).

[45] Chengzhang Zhu, Longbing Cao, and Jianping Yin, 'Unsupervised heterogeneous coupling learning for categorical representation', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **44**(1), 533–549, (2020).