BRNN-GAN: Generative Adversarial Networks with Bi-directional Recurrent Neural Networks for Multivariate Time Series Imputation

Zejun Wu*, Chao Ma*, Xiaochuan Shi*, Libing Wu*, Dian Zhang*, Yutian Tang† and Milos Stojmenovic‡
*Wuhan University, China, {zejunwu, chaoma, shixiaochuan, wu, zhangdian}@whu.edu.cn
†ShanghaiTech University, China, tangyt1@shanghaitech.edu.cn
‡Singidunum University, Serbia, mstojmenovic@singidunum.ac.rs

***Abstract*—Missing values appearing in multivariate time series often prevent further and in-depth analysis in real-world applications. To handle those missing values, advanced multivariate time series imputation methods are expected to (1) consider bi-directional temporal correlations, (2) model cross-variable correlations, and (3) approximate original data's distribution. However, most of existing approaches are not able to meet all the three above-mentioned requirements. Drawing on advances in machine learning, we propose BRNN-GAN, a generative adversarial network with bi-directional RNN cells. The BRNN cell is designed to model bi-directional temporal and cross-variable correlations, and the GAN architecture is employed to learn original data's distribution. By conducting comprehensive experiments on two public datasets, the experimental results show that our proposed BRNN-GAN outperforms all the baselines in terms of achieving the lowest Mean Absolute Error (MAE).**

***Index Terms*—multivariate time series, bi-directional RNN, GAN, missing value imputation**

I. Introduction

Multivariate time series are widely utilized in financial marketing [1], healthcare analytics [2], meteorology [3], traffic engineering [4] and many other application areas [34]. Time series can be treated as signals for multiple downstream tasks such as classification or regression in various applications. However, unexpected accidents, such as equipment failures, defective collection processes, or human mistakes [7], often lead to missing values in real-world multivariate time series , which hinders further analysis and applications of multivariate time series. To process incomplete multivariate time series, accurately imputing missing values is of the high priority. Otherwise, most of the downstream tasks might perform very poorly. Hence, to fully mine the potential of multivariate time series in real-world scenarios, missing value imputation becomes vital.

In general, there are three categories of missing values imputation methods. The first category usually contains case deletion methods [18], [19], which simply discards incomplete observations. Obviously, such methods might unconsciously abandon some important information, especially when the missing rate is high. Statistical methods such as mean imputation belong to the second category. They work well in time efficiency but can not achieve high imputation accuracy. Methods in the third category mainly rely on machine learning based models [5]. Recent machine learning based state-of-the-art methods include BRITS [6], an RNN-based method, and a GAN-based method named E^2GAN [7]. BRITS has excellent capability to model cross-variable correlations and bi-directional temporal correlations. E^2GAN takes advantage of a denoising auto-encoder and is an end-to-end model. However, BRITS fails to impute missing values with similar distribution to that of the original data, and E^2GAN ignores bi-directional correlations and cross-variable correlations.

Intuitively, taking bi-directional correlations and cross-variable correlations lead to a better imputation accuracy. It has been proved according to the study in [6], [13]. In addition, the importance of learning original missing time series' distribution has been emphasized in [16]. Our proposed BRNN-GAN models bi-directional correlations and cross-variable correlations while generating imputedtime series that approximate distribution of original observations. In Section IV, the impact of these three factors is validated via experiments. To learn bi-directional correlations, BRNN-GAN employs a bi-directional recurrent network which combines bi-directional imputed values on the basis of a temporal decay assumption. In the temporal decay assumption, historical influence of the past or future observations decays if the current value is missed for a long time [6]. Since GAN has shown great potential in distribution modeling [32], [33], it is adopted by our BRNN-GAN to capture the distribution of original data.

The main contributions of our work are summarized as follows:

(1) By adopting the GAN architecture, we propose BRNN-GAN, which models bi-directional correlations and cross-variable correlations while capturing original time series' distribution. Experimental results on two public datasets show the outstanding performance of BRNN-GAN for imputing missing values in multivariate time series.

(2) Furthermore, the impact of three key factors (i.e. bi-directional temporal correlation, cross-variable correlation and original data distribution approximation) is quantitatively studied.

The rest of this paper is organized as follows. The problem of missing value imputation in multivariate time series is formulated in Section II. In Section III, our proposed BRNN-

Zejun Wu and Chao Ma contribute equally to this work.

2690-5965/21/$31.00 ©2021 IEEE
DOI 10.1109/ICPADS53394.2021.00033

GAN is illustrated in detail. In Section IV, experimental results are provided with detailed analysis. Related work about missing value imputation of multivariate time series is discussed in Section V. Conclusions are made in Section VI to summarize our work in this paper.

II. Problem Formulation

A multivariate time series X = $\{x_1, x_2, \ldots, x_T\} \in \mathbb{R}^{T\times D}$ is denoted as a sequence of T observations. The t-th observation $x_t \in \mathbb{R}^D$ is composed by D variables $\{x_t^1, x_t^2, \ldots, x_t^D\}$, and was observed at timestamp s_t. The time gaps between different timestamps may not be the same because some values are missing.

To locate the missing values in x_t, a masking vector m_t, is introduced. The masking vector m_t is defined in (1).

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

The objective of missing value imputation in multivariate time series is to develop a method which is able to accurately estimate all missing values.

We evaluate the imputation performance in terms of *Mean Absolute Error*(MAE). Suppose that $label_t$ is the ground-truth of the t-th item, $pred_t$ is the predicted value of the t-th item, and there are N items in total. Then, MAE is defined in (2).

$$MAE = \frac{\sum_t \mid label_t - pred_t \mid}{N}. \tag{2}$$

The smaller the MAE is, the more accurate the imputation method is. In other word, the imputation method's objective could be defined as shown in (3).

$$\arg\min_F \frac{\sum_d \sum_t \mid (x_t^d - F(x_t^d))(1-m_t^d) \mid}{\mid\mid 1-m \mid\mid_1}, \tag{3}$$

where F is the imputation method function, whose inputs are original observations and outputs are the estimated values for those missing observations.

III. Methodology

A. Overall Architecture

BRITS [6] is a state-of-the-art method which captures both bi-directional and cross-variable correlations on the basis of bi-directional recurrent neural networks [9], [27]. We further enhances our model's ability to capture these correlations within time series by adopting a BRNN cell, which is an improvement on BRITS. The BRNN cell is illustrated in detail in Section III-B.

GAN has shown great potential in generating synthetic samples with a distribution similar to that of the training samples [16]. GAN consists of a Generator (G) and a Discriminator (D). The generator aims to generate imputed time series with the BRNN cell. The goal of the discriminator is to distinguish between fake generated samples and real original samples. The Wasserstein GAN (WGAN) makes some enhancements based on the original GAN and can improve learning stability and get away from the problem of mode collapse. Different from the original GAN, WGAN is formulated in (4) and (5).

$$L_{Generator} = \mathbb{E}_{z\sim P_g}\left[-D\left(G\left(z\right)\right)\right], \tag{4}$$

$$L_{Discriminator} = \mathbb{E}_{z\sim P_g}\left[D\left(G\left(z\right)\right)\right] - \mathbb{E}_{x\sim P_r}\left[D\left(x\right)\right]. \tag{5}$$

BRNN-GAN adopts WGAN to avoid mode collapse. The generator's input is original time series with missing values. The generator's output is the complete time series with all missing values imputed. Then, the imputed complete time series along with original incomplete time series is fed into the decoder, i.e., another BRNN cell.

Fig. 1 shows the overall architecture of BRNN-GAN. The trained generator can then be used to impute incomplete time series. At the imputation stage, the incomplete time series x is the input of the trained generator, and the trained generator outputs a generated time series $\widetilde{x}$. The generated values on unsuccessful observations in $\widetilde{x}$ can then be used to fill in the real but incomplete time series x. The imputation result $\overline{x}$ is computed as shown in (6).

$$\overline{x} = x \odot m + \widetilde{x} \odot (1-m). \tag{6}$$

In (6) $\odot$ represents the dot production operation.

B. Bi-directional RNN Cell

Fig. 2 shows the overall architecture of the BRNN cell.

Both the generator and the discriminator are implemented based on bi-directional RNN cells. The Bi-directional RNN cell (i.e. BRNN cell) makes improvements on BRITS [6]. The attractive property of the BRNN cell is its capability to model temporal correlations on both forward and backward time directions and to extract cross-variable correlations. Additionally, the BRNN cell decays the historical influence of the past or future observations if the current value is missed for a long time. The BRNN cell considers all correlations within the variable and across multiple variables.

Two matrices $\overrightarrow{\delta}$ and $\overleftarrow{\delta} \in \mathbb{R}^{T\times D}$ respectively record the time gap between current values and their corresponding last observed values in forward and backward directions. $\overrightarrow{\delta}_t^d$ in forward direction is defined as the time gap from the last observation to the current timestamp s_t. $\overrightarrow{\delta}_t^d$ is defined in (7).

$$\overrightarrow{\delta}_t^d = \begin{cases} s_t - s_{t-1} + \overrightarrow{\delta}_{t-1}^d,\ if\ t > 1,\ m_{t-1}^d = 0, \\ s_t - s_{t-1},\ if\ t > 1,\ m_{t-1}^d = 1, \\ 0,\ if\ t = 1 \end{cases} . \tag{7}$$

The backward $\overleftarrow{\delta}_t^d$ is defined in a similar way. We assume that temporal correlations decay proportionally to the time gaps.

$$\text{x} = \begin{bmatrix} / & / & 1 & 5 & \\ 4 & 2 & / & 7 & \ldots \\ 8 & / & / & 9 & \end{bmatrix}, \text{m} = \begin{bmatrix} 0 & 0 & 1 & 1 & \\ 1 & 1 & 0 & 1 & \ldots \\ 1 & 0 & 0 & 1 & \end{bmatrix}$$

The following part is an illustration of a 3-dimensional multivariate time series x, its corresponding m and its calculated example of $\overrightarrow{\delta}$.

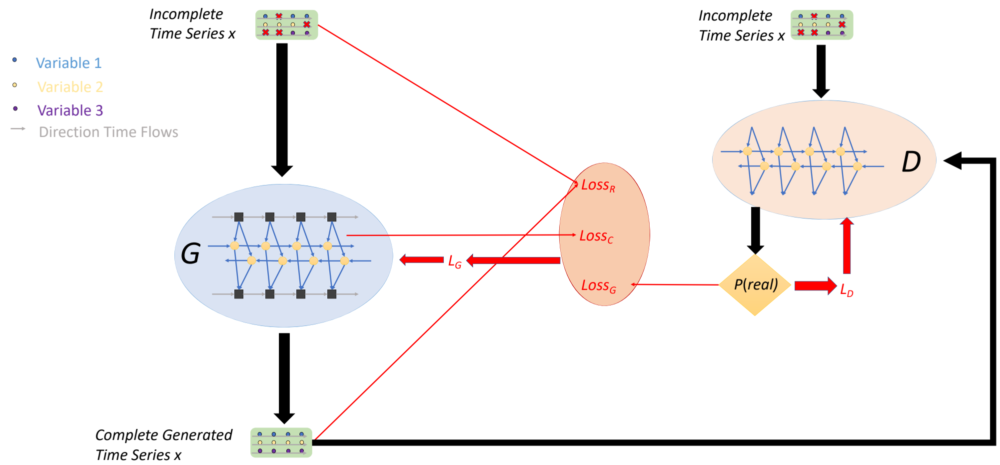

Fig. 1. The overall architecture of BRNN-GAN. The generator is composed by one bi-directional RNN cell and its input is original incomplete time series. The discriminator is composed by one bi-directional RNN cell as well and produces the probability of samples being real.

$$\overrightarrow{\delta}_t^d = \begin{bmatrix} / & / & 1 & 5 & \\ 4 & 2 & / & 7 & \dots \\ 8 & / & / & 9 & \end{bmatrix}.$$

As shown, time series x = { x_1, x_2, x_3, x_4 ... }. x_1 to x_4 are observed at $s_{1...4}$ = 0, 3, 5, 10 respectively.

We obtain BRNN cell's final output from the bi-directional recurrent layer. We define h_t^f as the hidden state from previous time steps in (9). The calculation of h_t^f takes advantages of γ_t^f, a factor that models the temporal decay influence based on the temporal decay assumption, as shown in (8).

$$\gamma_t^f = exp\left(-max\left(0, W_\gamma^f \overrightarrow{\delta}_t + b_\gamma^f\right)\right), \tag{8}$$

$$h_t^f = \sigma\left(W_h^f h_{t-1}^f \odot \gamma_t^f + U_h^f \overline{x}_t + b_h^f\right). \tag{9}$$

In (8) and (9), W_h^f, U_h^f, b_h^f, W_γ^f, b_γ^f are trainable parameters, and $\overline{x}_t$, the imputed values at timestamp s_t, are calculated using (6). σ is the sigmoid function.

Calculations in (8) and (9) to the forward direction are applied to the backward direction in the bidirectional recurrent layer for the calculation of h_t^b and γ_t^b, similarly. According to the temporal decay assumption, the bigger the time gap $\overrightarrow{\delta}_t$ (or $\overleftarrow{\delta}_t$) is, the smaller the temporal decay factor γ_t^f (or γ_t^b) is in (8).

The output of the recurrent layer is inputted to a fully-connected layer, which is adopted by the BRNN cell to estimate the target variable. The output $\widetilde{c}_t^f$, the estimated values in the forward direction, are defined in (10).

$$\widetilde{c}_t^f = W_x^f h_{t-1}^f + b_x^f, \tag{10}$$

In (10), W_x^f and b_x^f are trainable parameters. The calculation of $\widetilde{c}_t^b$ in backward direction is similar to (10).

A highlight of our BRNN cell is that the BRNN cell additionally considers cross-variable correlations. A variable-based estimation is defined as $\widetilde{z}_t$ in (11).

$$\widetilde{z}_t^f = W_z^f \overline{c}_t^f + b_z^f, \tag{11}$$

In (11), both W_z^f and b_z^f are corresponding trainable parameters. $\overline{c}_t^f$ is calculated using (6) with $\widetilde{x}$ replaced by $\widetilde{c}_t^f$. The diagonal elements of the parameter matrix W_z^f are restricted to be all zeros so that an estimation of one variable is based on other variables. In (12), $\beta_t^f \in [0,1]^D$ is used as the weight of combining the temporal-based estimation $\widetilde{c}_t^f$ and the variable-based estimation $\widetilde{z}_t^f$. In the forward direction, the forward combined vector is denoted as $\widetilde{x}_t^f$, defined in (13).

$$\beta_t^f = \sigma\left(W_\beta^f \left[\gamma_t^f \circ m_t\right] + b_\beta^f\right), \tag{12}$$

$$\widetilde{x}_t^f = \beta_t^f \odot \widetilde{z}_t^f + (1-\beta_t^f) \odot \widetilde{c}_t^f, \tag{13}$$

where W_β^f and b_β^f are trainable parameters. Here, $\circ$ indicates the concatenate operation. The backward combined vector $\widetilde{x}_t^b$ is calculated in the same way.

The last step is to combine imputed values of forward and backward directions based on the temporal decay influence assumption. λ_t^f and $\lambda_t^b \in [0,1]^D$ are data-driven combination factors [23] trained as model parameters. The combination factor in forward is defined in (14), and the combination factor in backward, λ_t^b, is calculated similarly. In this way, the final generated values are calculated in (15).

$$\lambda_t^f = exp\left(-max\left(0, W_\lambda^f \overrightarrow{\delta}_t + b_\lambda^f\right)\right), \tag{14}$$

$$\widetilde{x}_t = \lambda_t^f \widetilde{x}_t^f + \lambda_t^b \widetilde{x}_t^b, \tag{15}$$

where W_λ^f and b_λ^f are trainable parameters.

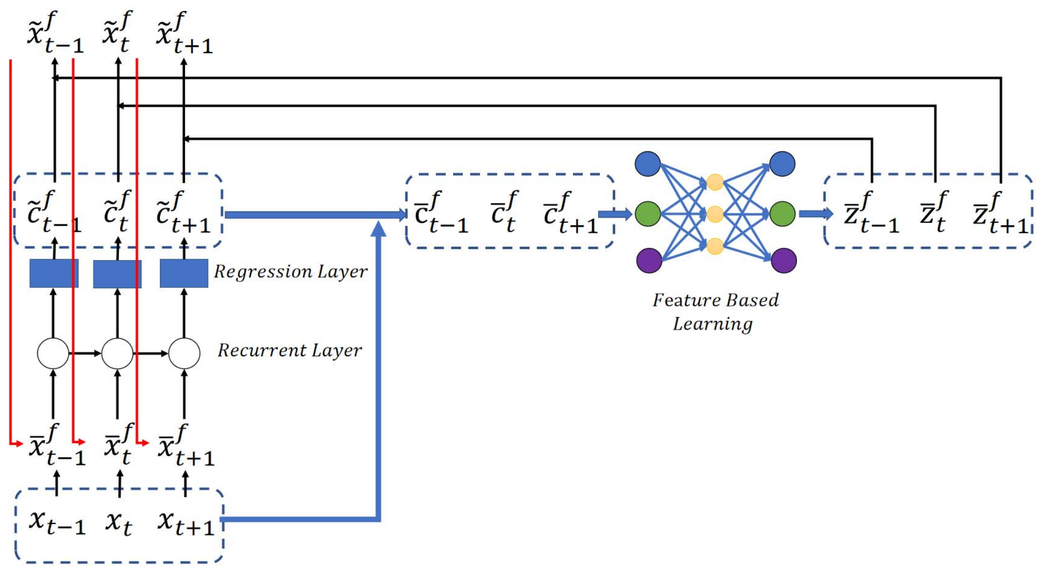

Fig. 2. The architecture of BRNN cell on forward direction. The architecture of BRNN cell on backward direction is similar to that on forward direction. The two network structures on forward and backward directions consist a BRNN cell.

C. Generator and Discriminator Architecture

We define a reconstruction loss $Loss_R$, a generative loss $Loss_G$ and a consistency loss $Loss_C$ in (16), (17) and (18). In (16), x refers to the original incomplete time series, and $G(z)$ equals to $\widetilde{x}$ in values.

$$Loss_R = || \, x \odot m - G(z) \odot m \, ||_2, \tag{16}$$

$$Loss_G = -D\left(\widetilde{x}\right), \tag{17}$$

$$Loss_C = Discrepancy\left(\widetilde{c}_t^f, \widetilde{c}_t^b\right), \tag{18}$$

where the mean absolute error is also used for calculating the discrepancy in our experiment. The loss function L_G of our Generator is defined in (19).

$$L_G = kLoss_R + Loss_G + Loss_C, \tag{19}$$

where k is a hyper-parameter controlling the discriminative loss and the squared error loss. In this way, the trained generator would be able to impute missing values by generating complete time series.

The discriminator consists of a BRNN cell and a regression layer. BRNN-GAN's discriminator is to distinguish between original real time series x and generated fake time series $\widetilde{x}$. The output of the discriminator is a probability that the input is a real sample. The loss function of the discriminator L_D is defined in (20).

$$L_D = -D\left(x\right) + D\left(\widetilde{x}\right). \tag{20}$$

IV. Evaluation

In Section IV-A, we firstly describe the selected baselines followed with the introduction to two public datasets for evaluation. In Section IV-B, the proposed BRNN-GAN is evaluated and compared with the baselines.

A. Baselines and Datasets

We select six baselines for validating the effectiveness of the proposed BRNN-GAN. All selected baselines are briefly described as follows:

• **MEAN**: The missing values are simply imputed with mean values.

• **IterativeSVD** [20] : Matrix completion by iterative low-rank SVD decomposition.

• **MICE** [10] : Multiple Imputation by Chained Equations (MICE) is a widely used imputation method which fills the missing values by creating multiple imputations with chained equations.

• **EM** [21] : An imputation method based on Expectation Maximization.

• **BRITS** [6]: This method is one of the state-of-the-art methods, which utilizes bidirectional recurrent networks to impute missing values in time series.

• **E^{2}GAN** [7]: Another state-of-the-art imputation method, which implements an end-to-end architecture to impute multivariate time series with missing values.

Two real-world datasets, including a meteorologic dataset and a medical dataset, are selected for the evaluation of our proposed BRNN-GAN.

Air Quality – The air quality dataset comes from an open kaggle dataset [35] that records Air Quality Index (AQI) and hourly data across stations and cities in India. This dataset consists of data from 2015 to 2020 in 26 cities in India. We use air quality data in Visakhapatnam in 2019 for evaluation. The AQI Index contains 13 variables, which are PM2.5, PM10, NO, NO2, NOx, NH3, CO, SO2, O3, Benzene, Toluene, Xylene and AQI. For evaluation, one observation for each variable is randomly eliminated and then used as the ground-truth. Data from the 3^{rd}, 6^{th} and 9^{th} months are selected as the test set. The validation set consists of data in December. The rest data is the train set.

Healthcare – The healthcare dataset is a public electronic medical record dataset that comes from the PhysioNet Challenge 2012 [17]. This dataset consists of records from 4,000 patients in intensive care unit (ICU). The data collected from each patient is a 48-hour multivariate incomplete time series that includes 35 variables such as age, weight, albumin, heart-rate, glucose, etc. The dataset is extremely sparse, with up to 78% missing values in total. We randomly eliminate an observation for each variable from data and use these eliminated observations as the ground-truth. We pre-processed this dataset via normalization. We randomly choose 70% samples for the train set, 20% samples for the test set, and the rest for the validation set.

We use the SGD algorithm to train BRNN-GAN. For BRNN-GAN, BRITS and E^2GAN, the hidden sizes are set to be 12 and 36 on Air Quality and Healthcare datasets, respectively. We adopt the strategy of updating 10 times for the generator and 1 time for the discriminator at one iteration [7].

We use the impyute package to implement MICE and EM. IterativeSVD is implemented based on the fancyimpute package [36]. BRITS is implemented on the basis of [6].

B. Experimental Results and Analysis

As shown in Table I, our proposed BRNN-GAN and all baselines are evaluated on Air Quality and Healthcare datasets in terms of the evaluation metric MAE. Since EM and MICE fail to impute missing time series on certain variables, we simply replace MAEs with mean values on these certain variables for evaluation. It is obviously observed that our proposed BRNN-GAN significantly outperforms all baselines by achieving the lowest MAE 25.5237 on Air Quality dataset and 0.5464 on Healthcare dataset, respectively. This validates the effectiveness of our proposed BRNN-GAN on the multivariate time series imputation task.

Fig. 3 shows the detailed experimental results of BRNN-GAN and all baselines on Healthcare dataset. Fig. 4 shows the detailed experimental results of BRNN-GAN and the baseline methods on Air Quality dataset. Fig. 5 shows the detailed experimental results of component impact analysis on Healthcare dataset. Fig. 6 shows the detailed experimental results of component impact analysis on Air Quality dataset.

To quantitatively measure the impact of the three factors mentioned in the Abstract, methods implemented by different combinations are tested and the results are shown in Table II. Method 1 is BRNN-GAN without variable-based estimation, which is unable to model cross-variable correlations. Method 2 and Method 3 do not model bi-directional temporal correlations. They adopt the uni-directional RNN cells instead of bi-directional ones. In method 4, the GAN architecture is removed, which weakens the model's ability to capture the data distribution.

By comparing the results of methods 1-4 with those of method 5 (i.e. BRNN-GAN), the significance of the three factors fully considered by our proposed BRNN-GAN is clearly identified. None of the three methods outperforms our BRNN-GAN. All in all, according to the experimental results and analysis, the effectiveness of the proposed BRNN-GAN is firmly validated with the impact of each factor quantitatively measured.

TABLE I
IMPUTATION PERFORMANCE COMPARISON IN TERMS OF MAE

Methods	Air Quality	Healthcare
MEAN	31.6729	0.6887
IterativeSVD	33.7336	0.5616
MICE	35.3778	1.4337
EM	61.7435	1.0456
BRITS	26.9370	0.5554
E^2GAN	26.7532	0.6326
BRNN-GAN	**25.5237**	**0.5464**

TABLE II
COMPONENT IMPACT ANALYSIS IN TERMS OF MAE

No.	Method	Air Quality	Healthcare
1	BRNN-GAN without variable-based estimation	27.3805	0.5923
2	BRNN-GAN with bi-directional RNN cells replaced by uni-directional RNN cells (forward)	26.1576	0.6536
3	BRNN-GAN with bi-directional RNN cells replaced by uni-directional RNN cells (backward)	27.0824	0.6368
4	BRNN-GAN that removes the architecture of GAN	26.9174	0.6472
5	BRNN-GAN	**25.5237**	**0.5464**

V. RELATED WORK

A substantial amount of research has focused on developing methods to handle missing values in time series. Due to the page limitation, only closely related methods are introduced in this section.

Statistical methods perform well in time efficiency. Such methods try to fill missing values with statistic variables such as mean values [22], last observed values [24] and mode values [25]. However, very few of statistical methods are able to achieve high imputation accuracy.

Nowadays, machine learning based methods have been widely used in various fields [26], [28]–[30], including multivariate time series imputation [31]. K-Nearest Neighbor (KNN) [11] algorithm computes the mean values of k nearest neighbors to estimate missing values. Matrix Factorization (MF) [12] algorithm factorizes the incomplete multivariate time series into low-rank matrices and adopts the product of these two matrices to impute the missing values. Multivariate Imputation by Chained Equations (MICE) [10] fills the missing values using the iterative regression model. MICE

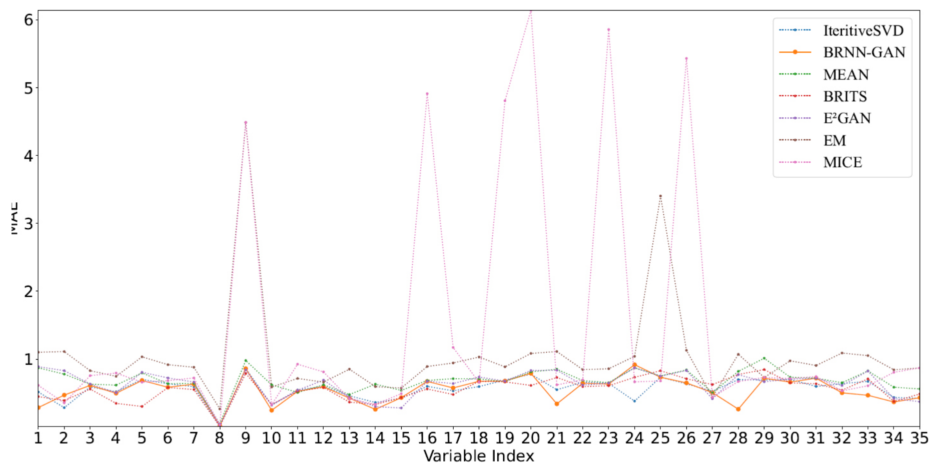

Fig. 3. The MAE results of BRNN-GAN and all baselines on each variable of Healthcare dataset.

imputes each incomplete variable by a separate model. IterativeSVD [20] is an imputation method robust to missing data. Expectation Maximization (EM) [21] conducts two steps (the E-step and the M-step) to impute missing values. The E-step calculates the expected complete data log likelihood ratio and the M-step searches for the parameters that maximize the log likelihood of the complete data. However, temporal correlations are not modeled by any of the above-mentioned methods.

To model temporal correlations, several imputation methods adopt recurrent neural networks. GRU-D [2] imputes missing values of a clinical dataset by jointly analyzing the last observed value and mean value to mine missing patterns of incomplete time series. M-RNN [13] models bi-directional temporal correlations using recurrent neural networks and operates across variables in addition. BRITS [6] is a novel neural network that models bi-directional and cross-variable correlations based on Bi-directional RNN. It directly estimates the missing values by a bi-directional recurrent dynamical system without any specific assumption. Unfortunately, these methods do not approximate original data's distribution.

Recently, Generative Adversarial Networks (GAN) [8], which aims to generate synthetic samples that obey the distribution of training dataset, has been utilized to impute missing values. GAIN [14] , a GAN-based imputation method, has made tremendous advances in data imputation. However, it does not show satisfactory performance on the time series imputation task [7]. Bi-GAN [23] is proposed for time series imputation and prediction, and it fails to model cross-variable correlations when applied on multivariate time series. A two-stage GAN-based time series imputation method [15] suffers from heavy time cost for model training. And the trained input

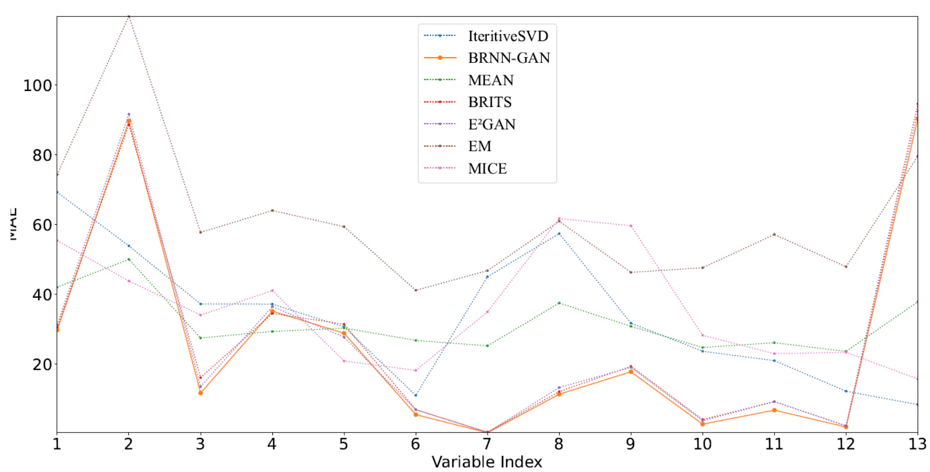

Fig. 4. The MAE results of BRNN-GAN and all baselines on each variable of Air Quality dataset.

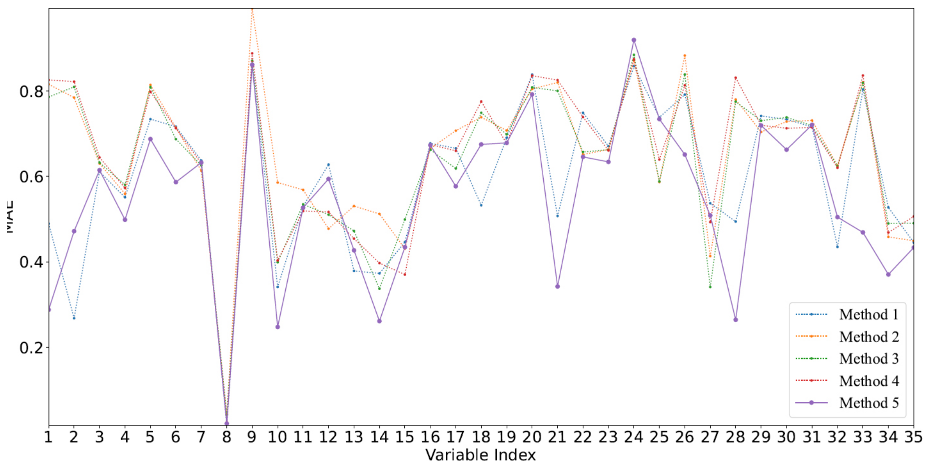

Fig. 5. The MAE results of each variable of component impact analysis on Healthcare dataset.

vectors of its model are not guaranteed to be optimal. E^2GAN [7] is an improvement on that two-stage method and shows better time efficiency and performance. However, E^2GAN does not model bi-directional correlations and cross-variable correlations.

From our perspective, an ideal time series imputation method is expected to consider bi-directional correlations and cross-variable correlations, while capture original data's distribution. This motivates the design of our proposed BRNN-GAN, which significantly outperforms state-of-the-art methods by taking all three key factors into account.

VI. Conclusion

In this paper, to accurately impute missing values in multivariate time series, we propose BRNN-GAN, a GAN model with Bi-directional recurrent neural networks. Three key factors are identified and integrated into the implementation of BRNN-GAN. By conducting comprehensive experiments on two public datasets Air Quality and Healthcare, the effectiveness of the proposed BRNN-GAN is firmly validated. Moreover, the impact of three key factors on imputation accuracy is quantitatively measured, which offers valuable insights for designing more effective imputation models for multivariate time series.

Acknowledgment

This work is supported by Wuhan University's teaching research project ”The innovation of contents construction and organization format of AI laboratory courses” under the category YB-10.

References

[1] B. Schölkopf and J. Peters, “The arrow of time in multivariate time serie,” *JMLR.org*, 2014.

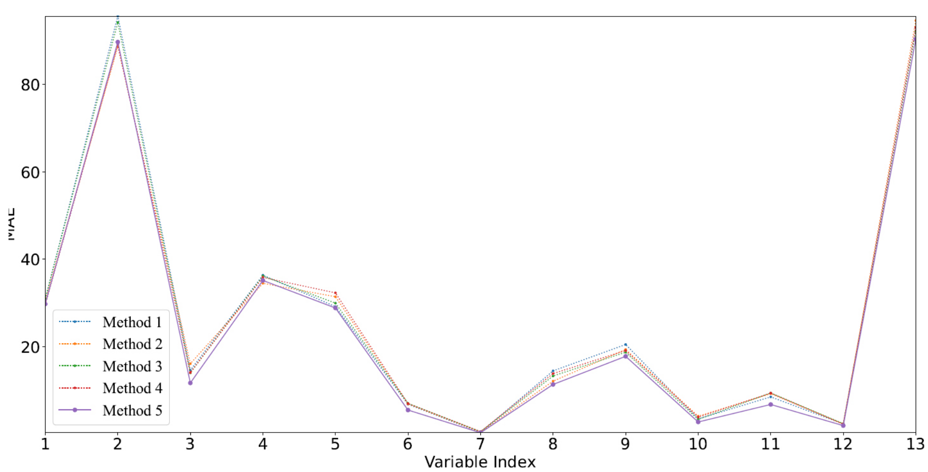

Fig. 6. The MAE results of each variable of component impact analysis on Air Quality dataset.

[2] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, “Recurrent neural networks for multivariate time series with missing values,” *Scientific Reports*, vol. 8, no. 1, p. 6085, 2018.
[3] S. Rani and G. Sikka, “Recent techniques of clustering of time series data: A survey,” *International Journal of Computer Applications*, vol. 52, pp. 1–9, 2012.
[4] J. Zhang, Z. Yu, and D. Qi, “Deep spatio-temporal residual networks for citywide crowd flows prediction,” 2016.
[5] G. Batista and M. C. Monard, “An analysis of four missing data treatment methods for supervised learning,” *Applied Artificial Intelligence*, 2003.
[6] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, and Y. Li, “Brits: Bidirectional recurrent imputation for time series,” in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018.
[7] Y. Luo, Y. Zhang, X. Cai, and X. Yuan, “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*. International Joint Conferences on Artificial Intelligence Organization, 7 2019, pp. 3094–3100.
[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014.
[9] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.
[10] I. R. White, P. Royston, and A. M. Wood, “Multiple imputation using chained equations: Issues and guidance for practice,” *Statistics in Medicine*, vol. 30, no. 4, pp. 377–399, 2011.
[11] A. T. Hudak, N. L. Crookston, J. S. Evans, D. E. Hall, and M. J. Falkowski, “Nearest neighbor imputation of species-level, plot-scale forest structure attributes from lidar data,” *Remote Sensing of Environment*, vol. 112, no. 5, pp. 2232–2245, 2008, earth Observations for Terrestrial Biodiversity and Ecosystems Special Issue.
[12] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup, “Scalable tensor factorizations with missing data,” in *Proceedings of the 2010 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, apr 2010.
[13] J. Yoon, “Multi-directional recurrent neural networks: A novel method for estimating missing data.”
[14] J. Yoon, J. Jordon, and M. Schaar, “Gain: Missing data imputation using generative adversarial nets,” 2018.
[15] Y. Luo, X. Cai, Y. Zhang, J. Xu, and X. Yuan, “Multivariate time series imputation with generative adversarial networks,” in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ser. NIPS’18. Red Hook, NY, USA: Curran Associates Inc., 2018, p. 1603–1614.
[16] Z. Guo, Y. Wan, and H. Ye, “A data imputation method for multivariate time series based on generative adversarial network,” *Neurocomputing*, vol. 360, pp. 185–197, 2019.
[17] I. Silva, G. Moody, D. J. Scott, L. A. Celi, and R. G. Mark, “Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012,” in *2012 Computing in Cardiology*, 2012, pp. 245–248.
[18] J. Kaiser, “Dealing with missing values in data,” *Journal of Systems Integration*, vol. 5, no. 1, pp. 42–51, 2014.
[19] L. O. Silva and L. E. Zárate, “A brief review of the main approaches for treatment of missing data,” *Intelligent Data Analysis*, vol. 18, no. 6, pp. 1177–1198, 2014.
[20] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman, “Missing value estimation methods for dna microarrays,” *Bioinformatics*, vol. 17, no. 6, pp. 520–525, 2001.
[21] G. E. Batista and M. C. Monard, “An analysis of four missing data treatment methods for supervised learning,” *Applied artificial intelligence*, vol. 17, no. 5-6, pp. 519–533, 2003.
[22] M. Kantardzic, *Data mining: concepts, models, methods, and algorithms*. John Wiley & Sons, 2011.
[23] M. Gupta and R. Beheshti, “Time-series imputation and prediction with bi-directional generative adversarial networks,” *arXiv preprint arXiv:2009.08900*, 2020.
[24] M. Amiri and R. Jensen, “Missing data imputation using fuzzy-rough methods,” *Neurocomputing*, vol. 205, pp. 152–164, 2016.
[25] A. Purwar and S. K. Singh, “Hybrid prediction model with missing value imputation for medical data,” *Expert Systems with Applications*, vol. 42, no. 13, pp. 5621–5631, 2015.
[26] L. Zhou, C. Ma, X. Shi, D. Zhang, W. Li, and L. Wu, “Salience-cam: Visual explanations from convolutional neural networks via salience score,” in *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–8.
[27] X. Wu, Z. Wu, and Y. Feng, “A text category detection and information extraction algorithm with deep learning,” *Journal of Physics: Conference Series*, vol. 1982, no. 1, p. 012047, jul 2021.
[28] Z. Li, C. Ma, X. Shi, D. Zhang, W. Li, and L. Wu, “Tsa-gan: A robust generative adversarial networks for time series augmentation,” in *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–8.
[29] Z. Song, D. Zhang, X. Shi, W. Li, C. Ma, and L. Wu, “Den-dql: Quick convergent deep q-learning with double exploration networks for news recommendation,” in *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–8.
[30] C. Ma, X. Shi, W. Li, and W. Zhu, “Edge4tsc: Binary distribution tree-enabled time series classification in edge environment,” *Sensors*, vol. 20, no. 7, 2020.
[31] C. Ma, X. Shi, W. Zhu, W. Li, X. Cui, and H. Gui, “An approach to time series classification using binary distribution tree,” in *2019 15th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)*, 2019, pp. 399–404.
[32] W. Li, L. Xu, Z. Liang, S. Wang, J. Cao, C. Ma, and X. Cui, “Sketch-then-edit generative adversarial network,” *Knowledge-Based Systems*, vol. 203, p. 106102, 2020.
[33] W. Li, L. Fan, Z. Wang, C. Ma, and X. Cui, “Tackling mode collapse in multi-generator gans with orthogonal vectors,” *Pattern Recognition*, vol. 110, p. 107646, 2021.
[34] W. Xu, C. Ma, S. Guo, and H. Zhou, “Efficient rate adaptation for 802.11af tvws vehicular access via deep learning,” in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.
[35] Vopani, “Air Quality Data in India (2015 - 2020),” 2020, https://www.kaggle.com/rohanrao/air-quality-data-in-india.
[36] A. Rubinsteyn and S. Feldman, “fancyimpute: An imputation library for python.” [Online]. Available: https://github.com/iskandr/fancyimpute