Building and Environment 215 (2022) 108896

Contents lists available at ScienceDirect

Building and Environment

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

Imputing missing indoor air quality data with inverse mapping generative adversarial network

Zejun Wu [a,1], Chao Ma [a,1], Xiaochuan Shi [a,*], Libing Wu [a], Yi Dong [b,c], Milos Stojmenovic [d]

[a] School of Cyber Science and Engineering, Wuhan University, Wuhan, China
[b] Institute of Rock and Soil Mechanics, Chinese Academy of Sciences, Wuhan, China
[c] Laboratory of Geomechanics and Geotechnical Engineering, Wuhan, China
[d] Department of Computer Science and Electrical Engineering, Singidunum University, Belgrade, Serbia

ARTICLE INFO

Keywords:
Indoor air quality
Missing data imputation
Generative adversarial network
Denoising auto-encoder
Bi-directional recurrent neural network

ABSTRACT

Sensors deployed all over the buildings are nowadays collecting a large amount of data, such as the Indoor Air Quality (IAQ) data which can provide valuable suggestions on improving indoor environments and energy consumption strategies. However, as treated as Multivariate Time Series (MTS), IAQ data often contain missing values that severely limit further analysis on them. Unfortunately, most of the existing methods fail to handle a couple of technical issues due to the complexity of MTS data, such as data distribution approximation, removing the redundancy, and so on. In this paper, we formulate the IAQ missing data imputation problem and propose an Inverse Mapping Generative Adversarial Network (IM-GAN) to tackle that problem. IM-GAN takes advantage of Bi-directional Recurrent Neural Network (BRNN), Denoising Auto-Encoder (DAE), and Generative Adversarial Network (GAN) to overcome the aforementioned technical issues. To validate the effectiveness of our proposed IM-GAN, we conduct comprehensive experiments on two public IAQ datasets GAMS and Gainesville. Results show that our IM-GAN achieves the new state-of-the-art performance in accurately estimating missing values in indoor air quality time series data, with the average performance of 0.1566 and 0.0789 in terms of *Mean Relative Error*, and 17.2884 and 2.7434 in terms of *Mean Absolute Error* on GAMS and Gainesville respectively at different missing rates. Our ablation study and visualization also validate that IM-GAN effectively overcomes the aforementioned technical issues by capturing data distribution, eliminating network saturation, and so on for IAQ data imputation.

1. Introduction

Buildings consume plenty of energy and cause the emission of "greenhouse" gases [1]. A number of studies have focused on analyzing building consumption data [2,3], such as indoor air quality (IAQ) data [4] (which is usually sensor data and can be processed as multivariate time series data) in order to foster energy efficiency strategies [5–7]. The approaches to the analyses of the data are generally classified as forward modeling and data-driven modeling [8,9]. Forward modeling relies on solid and complex engineering principles. The data-driven modeling process collected data using methods including computational intelligence and machine learning methods. Specifically, data-driven methods have been applied to the analysis of some building-related data successfully [10,11].

Both categories of methods, especially data-driven approaches, require the integrity of data for accurate further analysis [12]. However, building consumption data collected from the real world often contain missing values due to unexpected accidents, such as equipment failures, defective collection processes, or human errors [13]. These missing values pose significant challenges to the full use of data [14]. Two ways have been adopted to overcome the posed challenges. Firstly, implementing some system redundancy (e.g., adopting multiple loggers to monitor the same parameters) to prevent the loss of data during failures would possibly avoid such a problem; however, doing so would result in a much higher cost for equipment and maintenance [15]. Additionally, implementing the redundancy is not guaranteed to increase reliability for various reasons (e.g., complexity, human neglect) [16]. The second approach is to estimate the missing values and impute the data, which is more practical and promising. Hence, our work is aimed to design an imputation method that could be applied to IAQ data for accurately and effectively estimating missing values.

* Corresponding author.
E-mail addresses: zejunwu@whu.edu.cn (Z. Wu), chaoma@whu.edu.cn (C. Ma), shixiaochuan@whu.edu.cn (X. Shi), wu@whu.edu.cn (L. Wu), ydong@whrsm.ac.cn (Y. Dong), mstojmenovic@singidunum.ac.rs (M. Stojmenovic).
[1] Zejun Wu and Chao Ma contribute equally to this work.

https://doi.org/10.1016/j.buildenv.2022.108896
Received 12 December 2021; Received in revised form 3 February 2022; Accepted 10 February 2022
Available online 22 February 2022

So far, an overwhelming amount of methods have been proposed for imputing missing values in IAQ sensor data; however, most of them ignore some of the critical issues that should be considered to design effective and accurate imputation methods. (1) To be specific, the building-related sensor data is essentially temporal data; hence, some of the existing methods model the temporal correlations. However, the bi-directional temporal correlations (not only forward but also backward temporal correlations) should be investigated for an effective temporal correlation modeling, which has been issued only in a small number of studies [17,18]. Regrettably, even the already small number of studies are not aimed for applications on IAQ data imputation. (2) Since the comprehensive relationships between sensors may indicate important information, it is also important for imputation methods to take across-sensor correlations into consideration. (3) Furthermore, a number of methods focus on the criteria of imputation accuracy but fail to capture data distribution effectively [19] and thus impeding further use of imputed data, although they are seemingly effective by achieving low *Mean Absolute Error* (MAE) or *Mean Relative Error* (MRE), two evaluation criteria commonly used in imputation accuracy [17]. (4) The missing multivariate time series data is often highly redundant, so adopting modules such as Auto-Encoder (AE) helps find a robust representation that removes the noise (such as missing values) and redundancy in the design of good imputation methods [20,21].

Our work aims to handle a couple of complex technical issues in designing effective and accurate missing IAQ sensor data methods with the goal of facilitating further analysis on IAQ data. In this paper, a deep learning-based model, Inverse Mapping General Adversarial Network named IM-GAN, is proposed. IM-GAN employs a Denoising Auto-Encoder (DAE) to learn robust representation, which removes redundancy. The DAE consists of an encoder and a decoder, which perform mapping processes inverse to each other. A GAN structure is applied in our model to capture original data distribution. Moreover, bi-directional temporal correlations and across-sensor correlations are successfully modeled by our specially designed Bi-directional Recurrent Neural Network (BRNN) cell. Our proposed model IM-GAN is validated to be more effective in handling missing IAQ data than several state-of-the-art baselines under the multi-dimensional experiments.

In this work, our contributions are briefly summarized as follows:

(1) Four key technical issues (i.e. modeling variable correlations, modeling bi-directional temporal correlations, capturing data distribution, and removing redundancy) are identified for indoor air quality data imputation. We propose a novel method, IM-GAN, to handle these identified issues for missing indoor air quality data imputation.

(2)In our conducted comprehensive experiments, IM-GAN is validated to achieve the state-of-the-art performance on two public indoor air quality datasets at both data scale and variable scale in terms of *Mean Absolute Error* and *Mean Relative Error*. The impact of some key components in handling the identified issues is quantitatively studied in the ablation study.

(3) The complete multivariate time series data imputed by our proposed IM-GAN and state-of-the-art baselines are visually presented. The visualized results show that our IM-GAN overcomes a number of critical technical issues (e.g., failures in data distribution approximation, network saturation) more effectively than the baselines, which further validates our proposed method's effectiveness.

The rest of this paper is organized as follows. In Section 2, related works about missing value imputation are discussed from the traditional statistics-based perspective and the machine learning-based perspective. The problem of missing value imputation for IAQ data is formally formulated in Section 3. In Section 4, our proposed IM-GAN is illustrated in detail. In Section 5, afterward, comprehensive experiments are conducted to show the superiority of IM-GAN compared with state-of-the-art baselines. The conclusion is drawn in Section 6 to summarize our work in this paper.

2. Related work

There has been a substantial amount of research in developing methods for imputing data. Some closely related ones, particularly imputation methods for IAQ data, sensor data, or MTS data, are specifically discussed in this section.

2.1. Traditional statistics-based methods

Methods on building sensor data imputation proposed by 2016 are often based on simplified statistical methods (e.g., case deletion, mean value imputation, and zero value imputation [14]). Such methods perform well in time efficiency due to their concise mathematical adoption; however, they lead to poor reconstruction on those missing values and achieve unsatisfying imputation accuracy.

2.2. Machine learning-based methods

A large number of machine learning-based methods have been adopted in estimating missing values in IAQ and other time series data.

2.2.1. Non-deep learning-based methods

As one of the typical machine learning-based methods, Multivariate Imputation by Chained Equations (MICE) [22] fills the missing values using the iterative regression model. MICE imputes each incomplete variable by a separate model. The K-Nearest Neighbor (KNN) [23] algorithm computes the mean values of K nearest neighbors to estimate the missing values. Autoregressive models such as ARIMA [24], ARFIMA [25] and SARIMA [26] are also utilized to predict and impute missing values. A Matrix Factorization [27] algorithm factorizes the incomplete multivariate time series into two low-rank matrices and leverages the product of these two matrices to impute the missing values. Expectation Maximization (EM) [28–30] 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 followed by the M-step which searches for the parameters to maximize that log likelihood of the complete data. Also, a fault detection and diagnosis (FDD) method based on EM algorithm and Bayesian network (BN) was developed to fill in missing values in a chiller's dataset [31]. Other methods also include random forest (RF) based method [15] for the reconstruction of the average indoor air temperatures in a passive house. Regrettably, most of the Non-Deep Learning Methods fail to model the complex temporal correlations, which leads to unsatisfying imputation accuracy.

2.2.2. Deep learning-based methods

Deep learning-based methods have achieved great success in numerous fields [32–36], including data imputation. In existing studies [37,38], integrating auxiliary information will greatly help improve the performance of deep learning models.

To better utilize the temporal correlations, therefore, recurrent neural networks (RNNs) are integrated into a couple of machine learning and deep learning-based methods. For instance, GRU-D [39] smoothly imputes the missing values of a clinical dataset. Based on Gated Recurrent Unit (GRU), GRU-D jointly analyzes the last observed value along with the mean value to mine the missing patterns of incomplete time series. Rahman et al. [40] employ an RNN-based model for estimating missing values in electricity consumption data in commercial and residential buildings. A missing value imputation scheme based on Long short-term memory (LSTM) and transfer learning [41] is developed to insert electric consumption data collected in a campus lab building. M-RNN [18] tries to model correlations across and within variables by using bi-directional recurrent neural networks. BRITS [17] is a novel imputation method, which directly learns the missing values by developing a bidirectional recurrent dynamical system. However, even if such methods may effectively model bi-directional temporal correlations, they are not guaranteed to capture data distribution or

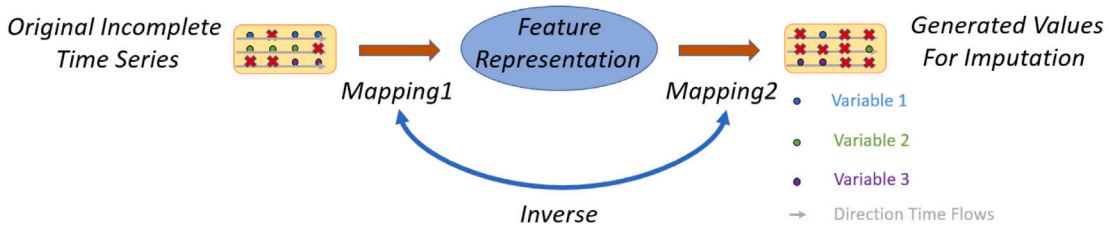

Fig. 1. An imputation task is understood as generating another incomplete time series out of a real incomplete time series through the inverse mapping processes.

some other critical technical issues as analyzed in the Introduction. We select BRITS, a state-of-the-art method employing recurrent neural networks as one of our baselines.

Recently, Generative Adversarial Networks (GANs) [42], which aim to generate synthetic samples that fall within the distribution of the training dataset, have been utilized to impute missing values. GAIN [43], a GAN-based imputation method, has made tremendous advances in data imputation; however, it does not show satisfactory performance when being applied to time series imputation tasks [13]. A two-stage GAN-based time series imputation method [44] requires substantial time for model training while its trained input vectors are not guaranteed to be optimal. MTS-GAN [20] adopts a GAN architecture specifically for MTS imputation. The results show that MTS-GAN performs robustly at different missing rates. Zhang et al. [45] employs the discriminator to force the generator to output imputed values that are close to the real ones during training. E^2GAN [13] shows better time efficiency and higher imputation accuracy than the previously associated two-stage GAN-based models [44]. However, most of these GAN-based methods do not consider bi-directional temporal correlations in their designs. Since E^2GAN [13] achieves the state-of-the-art performance, it is selected as one of the baseline methods.

To deal with the redundancy in data, previous works have applied auto-encoder (AE) to the field of missing data imputation with the goal of learning the hidden representations of real-world datasets with non-linear dependencies [46]. In [47], multivariate variational auto-encoders are applied on missing IAQ subway data imputation. A denoising stacked auto-encoder (DASE) is employed for traffic data imputation [48]. RDA [19] combines bidirectional recurrent neural networks with the denoising auto-encoder to finely model temporal correlations. RDA is one of the state-of-the-art methods and is selected as one of our baselines.

3. Problem formulation

The IAQ data in our study is inherently multivariate time series and can be represented in the form of multivariate time series. The problem of IAQ data imputation in our study is formulated in this section.

We denote a multivariate time series $X = \{ x_1, x_2, \ldots, x_T \} \in \mathbb{R}^{T\times D}$ as a sequence of T observations. The t-th observation $x_t \in \mathbb{R}^D$ is composed by D variables $\{ x_t^1, x_t^2, \ldots, x_t^D \}$. The timestamp that x_t was observed at is denoted as s_t. The time gaps between different timestamps may not be of equal duration.

Various accidents in the reality lead to missing values in x_t. To locate the missing values in x_t, a masking vector m_t is introduced, which is defined in Eq. (1).

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

Two matrices $f\delta$ and $b\delta \in \mathbb{R}^{T\times D}$ respectively record the time lags between current values and their corresponding last observed values in forward and backward directions. For the forward direction, $f\delta_t^d$ is defined as the time gap from the last observation to the current timestamp s_t. It is defined in Eq. (2).

$$f\delta_t^d = \begin{cases} s_t - s_{t-1} + f\delta_{t-1}^d, if\ t > 1, m_{t-1}^d = 0, \\ s_t - s_{t-1}, if\ t > 1, m_{t-1}^d = 1, \\ 0, if\ t = 1 \end{cases}. \tag{2}$$

The backward $b\delta_t^d$ is defined similarly. We assume that temporal correlations decay proportionally to the time gaps [17].

The objective of missing value imputation in multivariate time series is to develop a method so that all of the missing values can be accurately estimated. The imputation performance is evaluated in terms of *Mean Relative Error* (MRE) and *Mean Absolute Error* (MAE). MRE is defined in Eq. (3), where $label_t$ represents the ground-truth value of the t-th item, $pred_t$ is the imputation method's estimated value of the t-th item, and N is the number of items in total. MAE is defined in Eq. (4). The smaller the MRE and the MAE are, the more accurate the imputation is.

$$MRE = \frac{1}{N}\sum_t \frac{|\ label_t - pred_t\ |}{|\ label_t\ |}. \tag{3}$$

$$MAE = \frac{\sum_t |\ label_t - pred_t\ |}{N}. \tag{4}$$

4. Methodology

4.1. Overall architecture

We hypothesize that redundancy is harmful for missing value imputation in our IAQ data. Here, redundancy refers to information that is redundant, not applicable, or even harmful for IAQ data imputation. Hence, an intermediate robust feature representation, which is then used for generating values for imputation, should be extracted from the original data first. The intermediate representation is expected to contain robust features. The mapping from data to robust representation removes redundancy elegantly. This way, the imputation process can be regarded as two inverse mapping processes: mapping the real original data (processed as MTS in our applications on IAQ data) to a robust feature representation and inversely mapping the robust feature representation to generated values for imputation. Hence, an imputation task could be understood as generating another incomplete time series out of a real incomplete time series through inverse mapping processes, which is shown in Fig. 1.

IM-GAN employs a denoising auto-encoder to perform the inverse mapping task. The main idea of a denoising auto-encoder is to extract robust features by corrupting an encoder's input [49]. The encoding and corruption process and the decoding process are an inverse mapping pair. A denoising auto-encoder is excellent at generating another sample that is close to the original sample in values, which motivates the adoption of the denoising auto-encoder in IM-GAN.

However, a single denoising auto-encoder cannot guarantee that the distribution of its outputs (generated time series) approximates the original time series distribution [20]. GANs have been widely applied in various fields [50,51] and have shown great potential in generating synthetic samples with a distribution similar to that of the training samples [20]. A GAN is composed of a Generator (G) and a Discriminator (D), where the generator aims to map low-dimensional vectors (the feature representation in IM-GAN) to high-dimensional samples (values for imputation in IM-GAN). The discriminator's goal 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

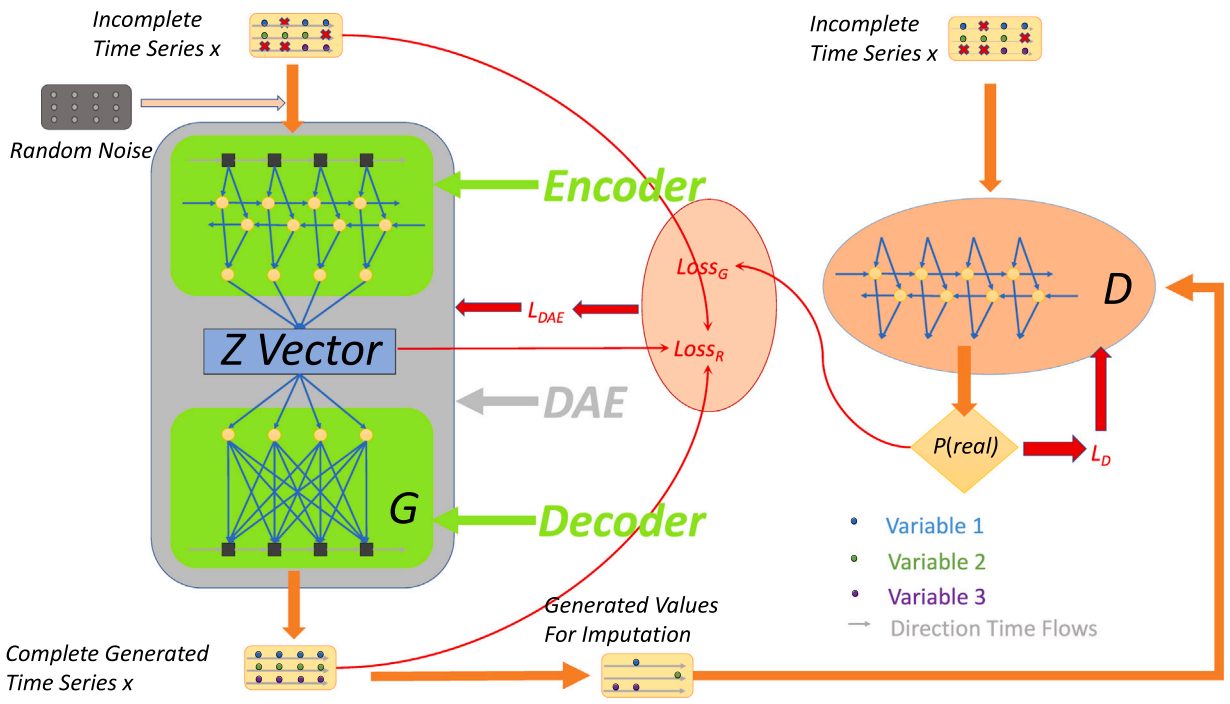

Fig. 2. The overall architecture of IM-GAN. The decoder of the denoising auto-encoder plays the role of the generator, which is composed of a regression layer. The discriminator consists of one bi-directional RNN cell and produces the probability of the samples being real.

get away from the problem of mode collapse [52]. Different from the original GAN, WGAN is formulated in Eq. (5) and Eq. (6).

$$L_G = \mathbb{E}_{z\sim P_g}\left[-D\left(G\left(z\right)\right)\right], \tag{5}$$

$$L_D = \mathbb{E}_{z\sim P_g}\left[D\left(G\left(z\right)\right)\right] - \mathbb{E}_{x\sim P_r}\left[D\left(x\right)\right]. \tag{6}$$

IM-GAN adopts WGAN to learn better inverse mappings and to avoid mode collapse. The encoder's input is a corrupted time series with missing elements. The decoder of the auto-encoder works as the IM-GAN's generator. In IM-GAN, the input of the decoder (the generator) is the output of the encoder, i.e., a low-dimensional vector z (the feature representation). We use the RMSProp algorithm to train our IM-GAN.

Fig. 2 shows the overall architecture of IM-GAN. The trained denoising auto-encoder can then be used to impute the incomplete time series. In the imputation stage, we first corrupt the incomplete time series x by adding random noise to make sure the DAE structure works similarly in the training stage as in the imputation stage. The trained denoising auto-encoder turns the corrupted, incomplete time series x into a generated time series $\widetilde{x}$. The generated values in $\widetilde{x}$ are then replaced by the actual values in the real time series x. The imputation result $\overline{x}$ is shown in Eq. (7), and $\odot$ is the dot product operation.

$$\overline{x} = x \odot m + \widetilde{x} \odot (1 - m). \tag{7}$$

4.2. Bi-directional RNN cell

Temporal information is of great importance and has been explored in various fields [53–55]. Modeling bi-directional temporal, as well as modeling variable correlations, is critical in the multivariate time series imputation task. A good modeling leads to better imputation results in previous work [17,18]. Both the denoising auto-encoder and the discriminator are implemented based on Bi-directional RNN cells. The Bi-directional RNN cell, BRNN cell, makes improvements on BRITS [17]. The attractive property of the BRNN cell is its enhanced capability to model temporal correlations in both forward and backward time directions and to extract 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 is utilized for better mapping by considering all correlations within the variable and across multiple variables.

BRNN cell's final output is obtained from the bi-directional recurrent layer. h_t^f (defined in Eq. (9)) is the hidden state from the previous time steps. γ_t^f (defined in Eq. (8)) represents the temporal decay factor, which is calculated using $f\delta_t$ and added to the calculation of h_t^f. σ is the sigmoid function.

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

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

where W_h^f, U_h^f, b_h^f, W_γ^f, b_γ^f are training parameters, and $\overline{x}_t$, the imputed values at timestamp s_t, are calculated using Eq. (7). h_t^b and γ_t^b in the backward direction are calculated similarly to the forward case (in Eq. (8) and Eq. (9)). Here, the bigger the time gap $f\delta_t$ (or $b\delta_t$) is, the smaller the temporal decay factor γ_t^f (or γ_t^b) is, which accords with the temporal decay assumption.

A fully-connected layer works as the regression component to generate the estimated values using values outputted by the hidden layer. The regression component's output $\widetilde{x}_t^f$, the generated values in the forward direction, are defined in Eq. (10).

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

where W_x^f and b_x^f are training parameters. $\widetilde{x}_t^b$ in the backward direction is calculated similarly to Eq. (10). The BRNN cell additionally considers variable correlations. A variable-based estimation is defined as $\widetilde{z}_t$ in Eq. (11).

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

where W_z^f and b_z^f are corresponding training parameters. $\overline{x}_t^f$ is calculated using Eq. (7) where $\widetilde{x}$ is replaced by $\widetilde{x}_t^f$. The diagonal elements of the parameter matrix W_z^f are restricted to zeros so that an estimation of one variable is based on other variables. In Eq. (12), $\beta_t^f \in [0,1]^D$ is used as a weight, which combines the variable-based estimation $\widetilde{z}_t^f$ and the temporal-based estimation $\widetilde{x}_t^f$. In the forward direction, the forward

combined vector $\widetilde{c}_t^f$ is denoted in Eq. (13).

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

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

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

The last step of the BRNN cell is to combine the imputed values of the forward and backward directions. λ_t^f and $\lambda_t^b \in [0,1]^D$ are data-driven factors trained as model parameters to combine estimations of forward and backward temporal directions based on the assumption of temporal decay influence. The factor for the forward direction is defined in Eq. (14), and the factor for the backward direction, λ_t^b, is calculated similarly. In this way, the final generated values are calculated in Eq. (15). W_λ^f and b_λ^f are parameters trained jointly with BRNN cell's other parameters.

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

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

4.3. The denoising auto-encoder and generator architecture

The architectures of the denoising auto-encoder and the generator are also shown in Fig. 2. To learn robust feature representations, DAE adopts the operations of dropping out original samples, adding random noise, and reconstructing complete samples [49]. The original samples are naturally missing, so there is no need to drop out the already missing samples' values. A random noise η that is sampled from a normal distribution $\mathcal{N}(0, 0.01)$ is added to the original incomplete time series in the encoding process. The time series with random noise, $x+\eta$, is the input of the BRNN cell. The output of the BRNN cell is a vector of the same length as the time series, $\ddot{x}$. The output is then mapped to a low-dimensional vector z and inversely mapped to a reconstructed vector of the same length as the time series, using two regression layers, respectively. Our decoder is the regression layer that maps vector z to the reconstructed vector (adopting BRNN cell as the decoder fails to improve imputation accuracy in our experimental validation, which accords with [19]). The reconstructed vector, which is the output, is the generated time series $\widetilde{x}$.

Here we define a reconstruction loss $Loss_R$ and a generative loss $Loss_G$ in Eq. (16) and Eq. (17).

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

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

where x here refers to the original incomplete time series that has not yet been modified by random noise, and $G(z)$ equals to $\widetilde{x}$. The loss function L_{DAE} of our denoising auto-encoder is defined in Eq. (18).

$$L_{DAE} = kLoss_R + Loss_G, \tag{18}$$

where k (set to be 1 in our implement) is a hyper-parameter controlling the discriminative loss and the reconstruction loss. This way, the trained denoising auto-encoder would be able to impute missing values by generating complete time series.

4.4. Discriminator architecture

The discriminator consists of bi-directional recurrent neural networks and their regression layers that output the probability. The final probability is obtained by calculating the mean value of the two temporal directions. Previous models (e.g., E^2GAN) feed the discriminator both the real time series x and the generated complete time series $\widetilde{x}$, while IM-GAN's discriminator distinguishes between the original real time series x and the generated incomplete time series $\widetilde{x} \odot (1-m)$. A generated complete time series contains original values and imputed values. This means that compared to a generated time series for imputation (incomplete), a generated complete time series is naturally more similar to its original time series in terms of data distribution. Hence, our design aims to enhance the performance to generate time series that approximate the original ones in terms of data distribution. The output of the discriminator is a probability that the input is a real sample. The encoder and the decoder are trained to be an inverse mapping pair. The loss function of the discriminator L_D is defined in Eq. (19). We adopt the training strategy of updating ten times for generator and one time for discriminator at one iteration in order to enhance the generator [13].

$$L_D = -D(x) + D\left(\widetilde{x} \odot (1-m)\right). \tag{19}$$

5. Experiments

In Section 5.1, we introduce two public datasets for evaluation. The selected baselines are described in 5.2. In Section 5.3, the proposed IM-GAN is evaluated and compared with the selected baselines.

5.1. Datasets and preprocessing

Two real-world indoor air quality public datasets are selected for the evaluation of the proposed IM-GAN and are described below.

GAMS – This dataset [56] consists of indoor and outdoor air quality data recorded by the gams Environmental Monitoring company (denoted as GAMS). The indoor data is recorded from GAMS indoor air quality monitor, and the outdoor data is tracked via GAMS API. The dataset consists of over 130,000 observations for each variable and six variables in total, which are CO_2, Humidity, PM10, PM2.5, Temperature, and Voc. The GAMS indoor data is used in our evaluation.

Gainesville – The dataset [57] was collected in a test building between 17 May 2020 (Sunday) and 18 September 2020 (Friday) in Gainesville city, Florida, United States. The building chosen for air quality monitoring is mechanically conditioned throughout the year, and the windows are closed to meet the American Society of Heating, Refrigerating, and Air-Conditioning Engineers (ASHRAE) Standard 62.1–2019 [58]. The standardized US EPA protocol for characterizing IAQ in large office buildings is followed to reduce measurement uncertainty [59,60]. A four-month indoor and outdoor air quality measurement was carried out with 10-minute sampling intervals for 24 h continuously. We only use the IAQ data for evaluation.

We select the first 60% percent data of each dataset in temporal sequence as the train set, the last 20% percent as the test set, and the rest as the validation set.

5.2. Baselines

The baselines selected for validating the effectiveness of the proposed IM-GAN are briefly described as follows:

- **MEAN**: The missing values are estimated as mean values.
- **EM (i.e. Expectation Maximization)**[28]: This is an Expectation Maximization-based imputation method and has been widely applied to missing data imputation of all categories of data, including IAQ data.
- **RDA** [19]: RDA is one of the state-of-the-art methods and combines an RNN with a denoising auto-encoder architecture for imputation.
- **BRITS** [17]: This method is one of the state-of-the-art methods. It utilizes bidirectional recurrent networks to impute time series and outperforms most of the existing imputation methods.
- **E^{2}GAN** [13]: This is one of the state-of-the-art methods. E^2GAN uses an end-to-end architecture to impute multivariate time series.

For EM, the parameter *eps* that decides the amount of minimum change between iterations to break (if relative change < *eps*, converge) is set to be 0.1. For deep learning based methods including

Table 1
Experimental settings.

Dataset	GAMS				Gainesville			
Methods	RDA	BRITS	E^2GAN	IM-GAN	RDA	BRITS	E^2GAN	IM-GAN
Epoch	200				200			
Batch Size	80				20			
Sample Length	20				20			
RNN Hidden Size	5, 10, 20, 40				5, 10, 20, 40			
Intermediate Vector Z Size	20	NA	20		20	NA	20	

RDA, BRITS, E^2GAN and our IM-GAN, which require a group of parameters, the details of our experiment settings are provided as shown in Table 1. To balance the expected computational time and imputation performance, we determine hyper-parameters based on the grid search according to our experiences, which reduces the expected hyper-parameter searching space. To make a fair comparison, we set the epoch number and batch size to be the same for all methods on the same dataset. A short sample length may limit the effects of temporal correlation modeling for RNN-based methods. The "infinite memory" of RNN-based methods is largely absent when applied to long-term time series in practice [61], suggesting that the sample length may not be set long. Following previous works [62], we set the sample lengths to be 20, an appropriate value neither too big nor too small. We set the Z size to be the same as the sample length. The RNN Hidden Size is selected using the grid search with the searching space presented. We implement all the RNN-based models with Pytorch 1.7.1 in Python 3.7.9. RNN-based Models are trained with GPU RTX 2080Ti. MEAN is also implemented with Python libraries. EM is implemented by utilizing the third-party Python library called impyute [63]. For the imputation accuracy experiments, since the output of the neural network is non-deterministic, the experiment with the same setting will be run for 10 iterations and the average is taken as the final performance.

5.3. Experimental results and analysis

5.3.1. Imputation accuracy on dataset scale

As shown in Table 2, all baselines, along with our proposed IM-GAN are evaluated on the two IAQ datasets: GAMS and Gainesville, in terms of the MRE and MAE metrics. MRE could more accurately measure the imputation accuracy than MAE since it uses relative values. For simplicity, we save four decimal places of the imputation metric values. It is obviously observed that our proposed IM-GAN significantly outperforms all baselines by achieving the lowest MRE on both datasets at all missing rates and the lowest MAE in most cases. For the average performance of different missing rates, our IM-GAN outperforms all the baselines in terms of both MRE and MAE. This shows the effectiveness of the proposed IM-GAN on the MTSI task in imputation accuracy. We credit the superiority of IM-GAN to its successfully handling the four key technical issues as we have analyzed.

5.3.2. Imputation accuracy on variable scale

We present the MRE accuracy of all single variables at different missing rates on both two datasets. The results are shown in Figs. 3 and 4. Though some baselines achieve the lowest MRE on certain variables, they impute rather inaccurately on other variables; however, it can be clearly observed that our IM-GAN achieves stable and excellent imputation performance at the variable scale in terms of the imputation accuracy metric MRE.

5.3.3. Impact analysis of model components

We further conduct an ablation study to validate the effectiveness of our model components. The results are shown in Table 3. Uni IM-GAN adopts a uni-directional recurrent neural network to replace our BRNN cell. Namely, Uni IM-GAN's final estimation values are the outputs

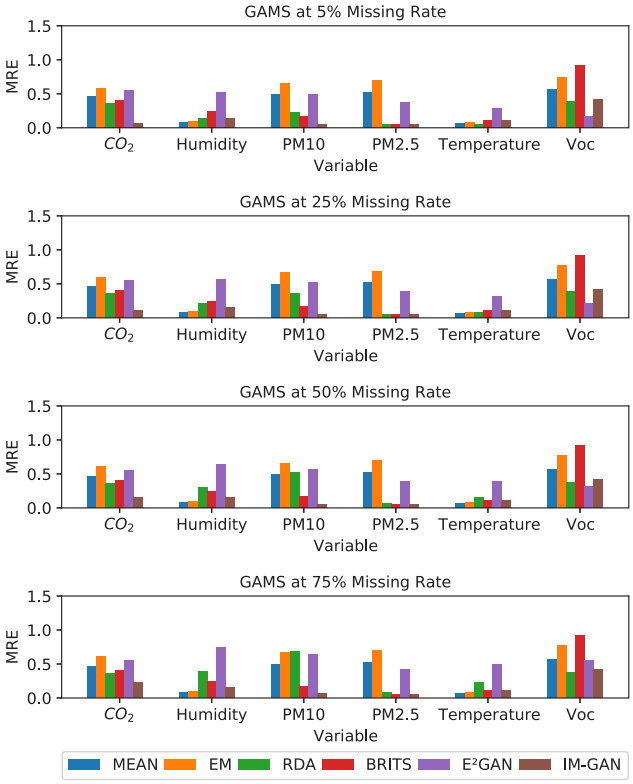

Fig. 3. Imputation Accuracy on Variable Scale in GAMS.

of Eq. (13) instead of those of Eq. (15). IM-GAN w.o. discriminator removes the discriminator of the IM-GAN. IM-GAN w.o. decoder removes the decoder of IM-GAN and instead uses the encoder of IM-GAN as its generator. The parameters of all the models in the ablation study are set to be the same as IM-GAN, as described in Table 1. In the ablation study, it can be concluded that the key components in our methods all contribute to improving the imputation accuracy.

5.3.4. Visualization

We use t-distributed Stochastic Neighbor Embedding (t-SNE) [64] to reduce the dimensionality of the large-scale time series data and visualize the compressed data in order to make a deep analysis of the values generated by imputation methods. As a variation of Stochastic Neighbor Embedding (SNE) [65], t-SNE works better in simultaneously preserving a dataset's local and global structure compared to its predecessor and other commonly used dimensionality reduction methods [66], such as Principal Component Analysis (PCA) [67].

In detail, for each variable, the data is divided into numeric time series with the length of 20, the exact length of samples for training the models in our experimental settings. Hence, each point in the visualization of the generated data represents a single sample outputted by imputation methods. Every time series with the length of 20 is

Table 2
Imputation performance of all methods in terms of MRE (MAE).

Dataset	GAMS				
Missing Rate	5%	25%	50%	75%	Average performance
MEAN	0.3629 (57.6769)	0.3623 (57.7004)	0.3632 (57.9098)	0.3634 (57.8966)	0.3629 (57.7959)
EM	0.4780 (72.5685)	0.4834 (74.3223)	0.4858 (75.0222)	0.4907 (75.4677)	0.4845 (74.3452)
RDA	0.1999 (43.1661)	0.2430 (44.2770)	0.2986 (45.7495)	0.3534 (47.1683)	0.2737 (45.0902)
BRITS	0.3163 (50.1891)	0.3161 (50.0978)	0.3160 (50.1216)	0.3161 (50.1279)	0.3161 (50.1341)
E^2GAN	0.3988 (71.7267)	0.4266 (72.1706)	0.4783 (73.1039)	0.5649 (74.4553)	0.4672 (72.8641)
IM-GAN	**0.1417 (9.5514)**	**0.1499 (13.9010)**	**0.1604 (19.6819)**	**0.1743 (27.7341)**	**0.1566 (17.2884)**
Dataset	Gainesville				
Missing Rate	5%	25%	50%	75%	Average Performance
MEAN	0.3543 (2.7325)	0.3510 (2.7910)	0.3582 (2.7743)	0.3560 (2.7820)	0.3549 (2.7700)
EM	0.5087 (3.6600)	0.5528 (3.9519)	0.5608 (3.8366)	0.5241 (3.8367)	0.5366 (3.8213)
RDA	0.0856 (1.8049)	0.1306 (3.0174)	0.1689 (4.3801)	0.2102 (5.8345)	0.1488 (3.7592)
BRITS	0.0921 (3.2190)	0.0933 (3.2171)	0.0941 (3.2403)	0.0945 (3.2382)	0.0935 (3.2286)
E^2GAN	0.2092 (9.2246)	0.2344 (10.2900)	0.3017 (12.2538)	0.4248 (15.6137)	0.2925 (11.8455)
IM-GAN	**0.0785** (2.7834)	**0.0773 (2.5479)**	**0.0767 (2.6811)**	**0.0831** (2.9610)	**0.0789 (2.7434)**

Table 3
Ablation Study of Imputation performance of all methods in terms of MRE (MAE).

Dataset	GAMS				
Missing Rate	5%	25%	50%	75%	Average performance
Uni IM-GAN	0.1676 (12.0508)	0.2083 (19.7351)	0.2627 (31.1548)	0.3168 (41.0672)	0.2388(26.0020)
IM-GAN w.o.discriminator	0.1831 (11.1135)	0.2448 (17.6943)	0.3059 (30.7687)	0.3604 (44.8183)	0.2735(26.0987)
IM-GAN w.o.decoder	0.2431 (55.3399)	0.2432 (55.3287)	0.2432 (55.3191)	0.2434 (55.3639)	0.2432 (55.3379)
IM-GAN	**0.1417 (9.5514)**	**0.1499 (13.9010)**	**0.1604 (19.6819)**	**0.1743 (27.7341)**	**0.1566 (17.2884)**
Dataset	Gainesville				
Missing Rate	5%	25%	50%	75%	Average performance
Uni IM-GAN	0.0816 (**2.6439**)	0.0968 (2.7182)	0.1001 (2.9027)	0.1147 (3.4584)	0.0983 (2.9308)
IM-GAN w.o.discriminator	0.1293 (4.9281)	0.1364 (5.4643)	0.1451 (6.0572)	0.1563 (6.5471)	0.1418 (5.7492)
IM-GAN w.o.decoder	0.1098 (2.7689)	0.1186 (3.0731)	0.1293 (3.6456)	0.1402 (4.2023)	0.1245 (3.4225)
IM-GAN	**0.0785** (2.7834)	**0.0773 (2.5479)**	**0.0767 (2.6811)**	**0.0831 (2.9610)**	**0.0789 (2.7434)**

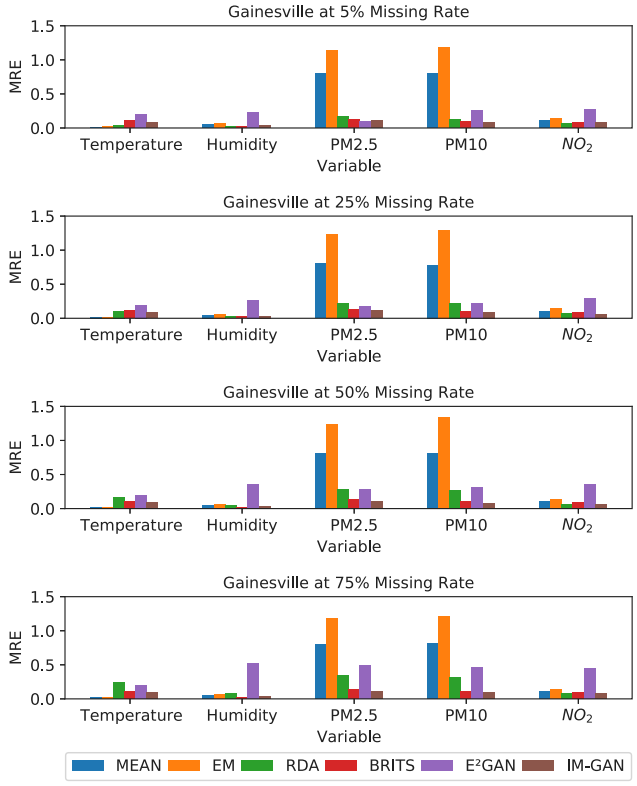

Fig. 4. Imputation Accuracy on Variable Scale in Gainesville.

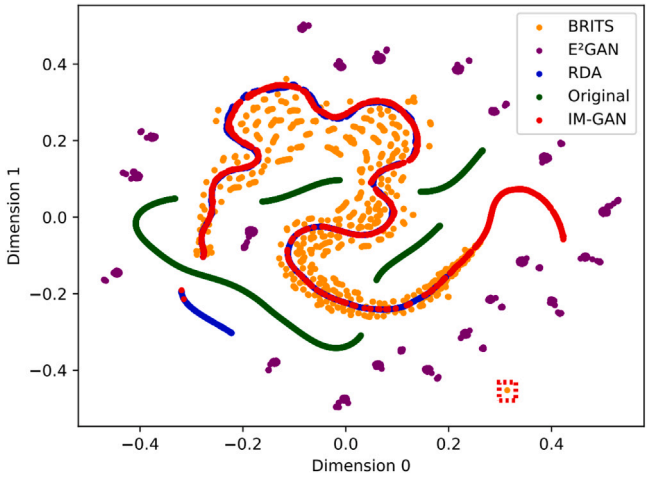

Fig. 5. Visualization of Generated and Original Values on CO_2 of GAMS.

further processed into a 2-dimensional vector. We visualize these 2-dimensional vectors obtained by processing original data and generated data by imputation methods, on every single variable separately, on the GAMS and Gainesville datasets. Since MEAN and EM do not generate values on successful observations during the imputation process, we only process original data and the data generated by BRITS, RDA, E^2GAN and our IM-GAN. Specifically, we choose data generated at the highest missing rate, 75% for Gainesville and 5% for GAMS. In general, the visualization results show that the baselines and the proposed IM-GAN work better on the Gainesville dataset and that IM-GAN work better than the baselines on both GAMS and Gainesville datasets.

Network Saturation

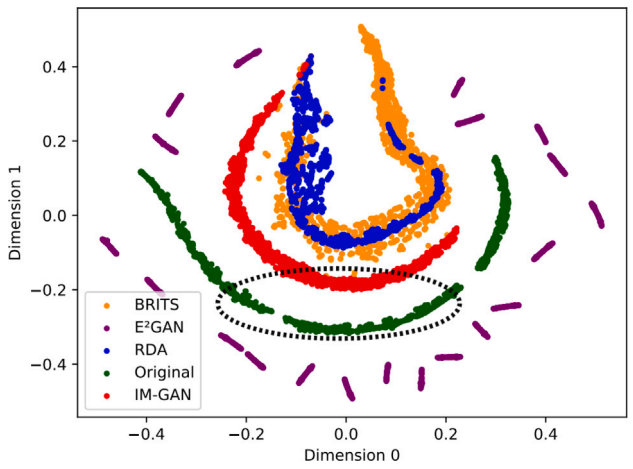

Fig. 6. Visualization of Generated and Original Values on Humidity of GAMS.

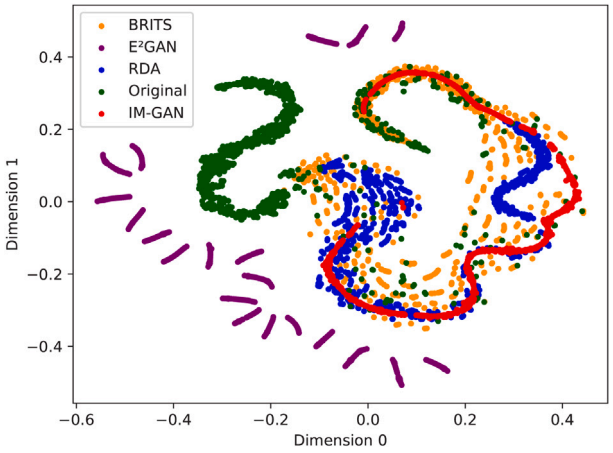

Fig. 9. Visualization of Generated and Original Values on Temperature of GAMS.

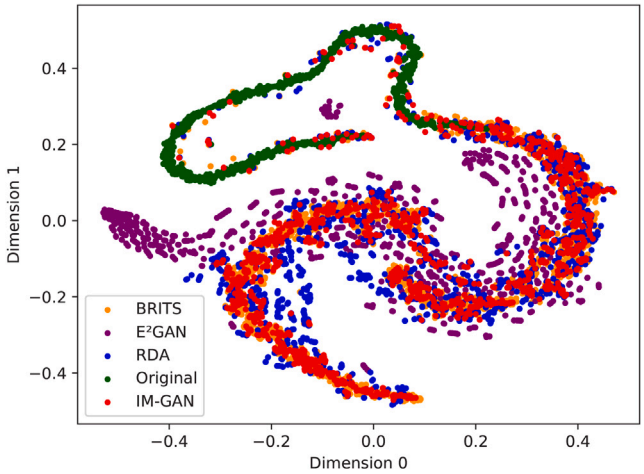

Fig. 7. Visualization of Generated and Original Values on PM10 of GAMS.

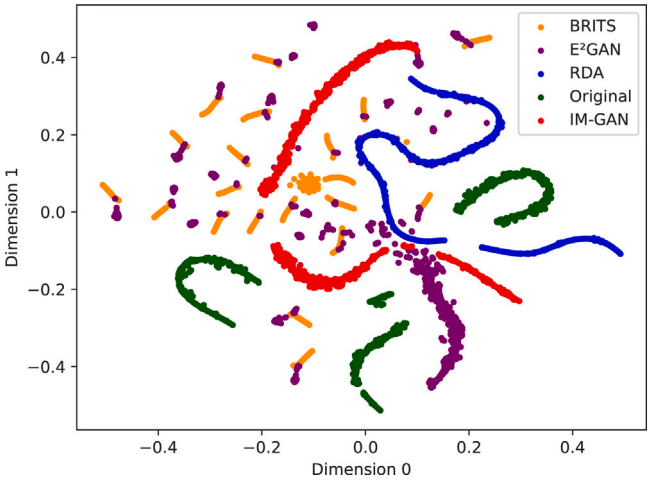

Fig. 10. Visualization of Generated and Original Values on Voc of GAMS.

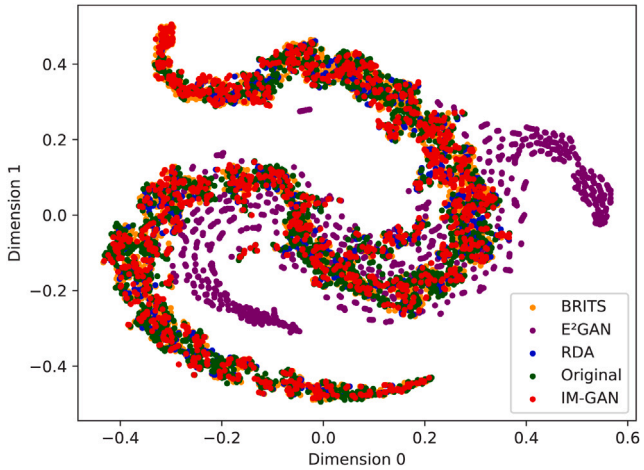

Fig. 8. Visualization of Generated and Original Values on PM2.5 of GAMS.

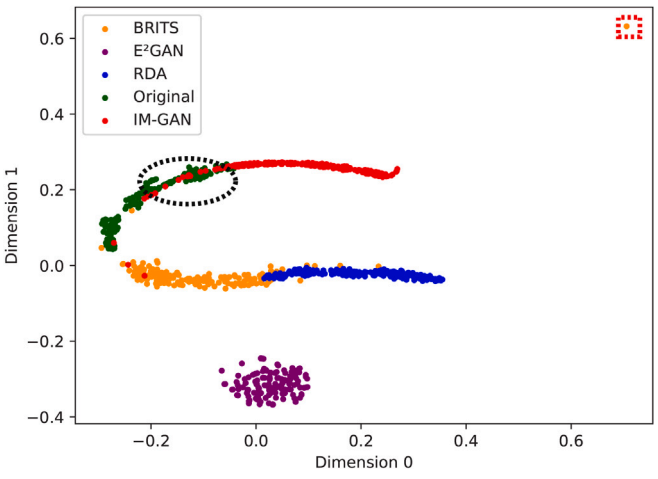

Fig. 11. Visualization of Generated and Original Values on Temperature of Gainesville.

Network saturation [68] was a major modeling complexity in the case of feed-forward models, including denoising auto-encoders. If the network saturation occurs, the model tends to produce similar outputs, even though the input sequences are actually of apparent differences. A criterion for network saturation, Saturation performance metric (SAT), has been proposed to identify indoor air quality data imputation models with saturation issues [62,69]. In the experiments, SAT is considered as the mean variance lower than 10% between every normalized generated 24-steps time-series. However, some original data inherently has a relatively low variance, which makes the criterion not applicable on

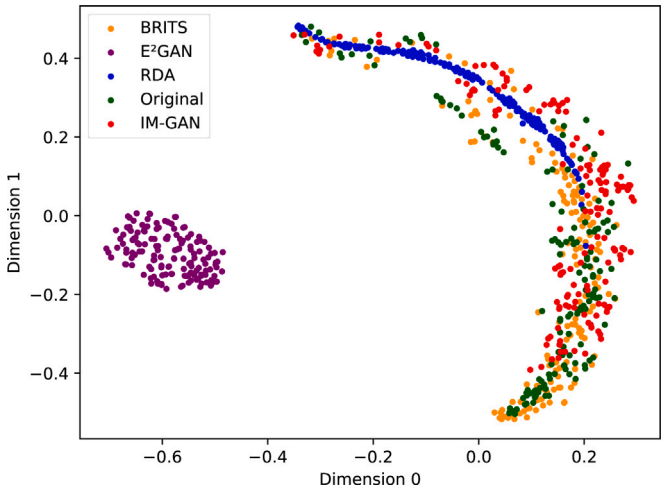

Fig. 12. Visualization of Generated and Original Values on Humidity of Gainesville.

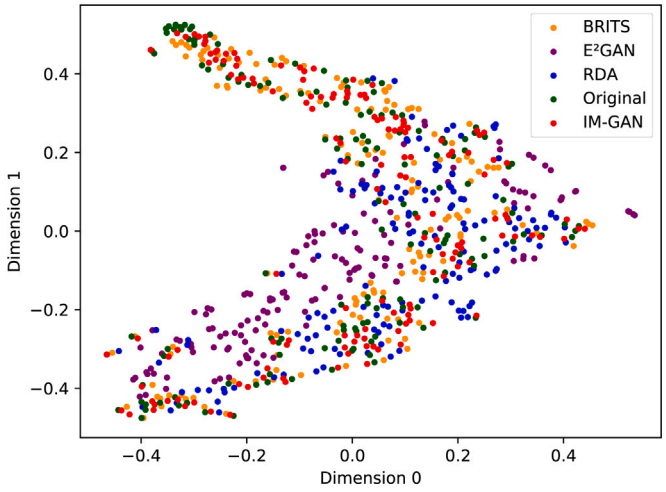

Fig. 13. Visualization of Generated and Original Values on PM2.5 of Gainesville.

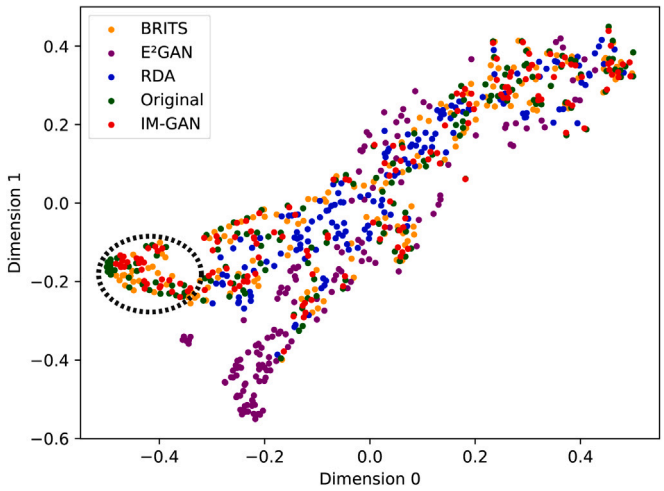

Fig. 14. Visualization of Generated and Original Values on PM10 of Gainesville.

all datasets. In our visualization, the phenomenon of samples clustering is an indication of network saturation, suggesting that the imputation method's outputs are very similar and do not follow temporal variation.

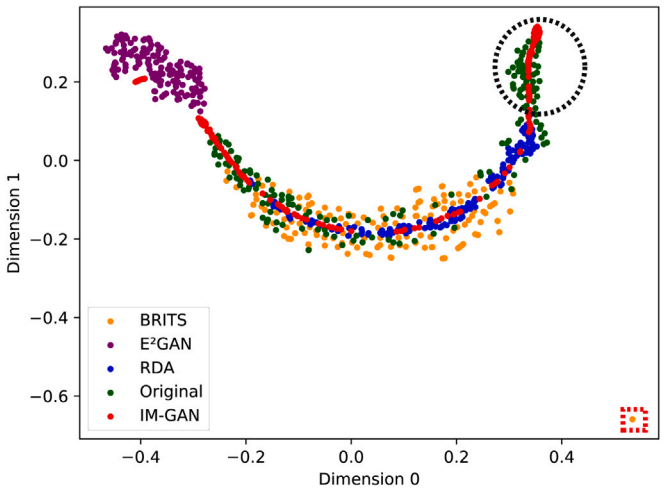

Fig. 15. Visualization of Generated and Original Values on NO_2 of Gainesville.

As shown in Figs. 5–15, the green samples, which are the original data, generally present in lines, since the samples accord with temporal correlations. This indicates that good samples generated by imputation methods should also present in lines if the problem of network saturation is overcome and the generated data captures the long-term temporal correlations. However, E^2GAN fails to present in lines on some variables. Figs. 5, 11, 12 and 15 are typical examples of the phenomenon, where all the other methods, including our IM-GAN present in lines and approximate the green original data while the purple dots, which represent the data generated by E^2GAN, tend to present in clusters. In our view, the AE network structure of E^2GAN is too complex (with the GRU-D cell employed in both the encoder and the decoder), possibly leading to network saturation in our experiments. Obviously, our IM-GAN gets rid of such problems on all variables.

Data Distribution

It has been theoretically suggested that adopting the GAN structure in an imputation model could enhance the model's ability to capture the data distribution. By analyzing the visualization results, we discover that some values generated by BRITS are exceptionally far from the normal range in data distribution marked as red dashed squares. As can be observed in Figs. 5, 11 and 15, some yellow sample points lie in the corner of the graph, far from those normal samples. In contrast, our IM-GAN gets rid of such problems by employing a GAN architecture and successfully generates synthetic samples that fall within the distribution of the original data.

Imputation Accuracy and Data Value Approximation

On the Gainesville dataset, RDA and BRITS are strong competitors to our proposed IM-GAN by both achieving low MRE and MAE on most of the variables and approximating well with the original values in the visualization. However, due to the BRNN cell's excellent ability to model temporal correlations based on the temporal decay assumption and the GAN's ability to generate samples that fall within the original samples' distribution, our IM-GAN still generates values that approximate the best with the original values. On the GAMS dataset, the imputation accuracy could be better improved for all methods, but our IM-GAN stands out from the competition. We mark some highlights of the IM-GAN's imputation results using the black dotted ovals.

6. Conclusion

A novel method, IM-GAN, is proposed for IAQ missing data imputation. Experiment results show that IM-GAN achieves the state-of-the-art imputation accuracy in both MRE and MAE on two public IAQ datasets on both dataset scale and variable scale. The ablation study validates

the effectiveness of the three key model components: the bi-directional networks, the GAN architecture, and the AE structure. The data generated by imputation methods are further visualized, with the results showing IM-GAN overcoming a series of critical challenges in IAQ data imputation, including network saturation, failures in data distribution approximation, and failures in data value approximation. Hopefully, IM-GAN could be further applied in some other building-related data imputation other than IAQ.

CRediT authorship contribution statement

Zejun Wu: Conceptualization, Methodology, Software, Writing – original draft. **Chao Ma:** Writing – review & editing, Validation, Supervision, Investigation, Conceptualization. **Xiaochuan Shi:** Data curation, Visualization, Writing – original draft. **Libing Wu:** Investigation, Formal analysis. **Yi Dong:** Project administration, Supervision. **Milos Stojmenovic:** Writing – review & editing.

Declaration of competing interest

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

Acknowledgments

This work is partially supported by the Key R&D Program of Hubei Province, China (No. 2021BAA039), the Humanities and Social Sciences of Ministry of Education Planning Fund, China (No. 21YJAZH073), and the National Natural Science Foundation of China (No. 72071211).

References

[1] EC, Directive (eu) 2018/844 of the european parliament and of the council amending directive 2010/31/eu on the energy performance of buildings and directive 2012/27/eu on energy efficiency, Off. J. Eur. Union 61 (2018) 75–76, URL: http://data.europa.eu/eli/dir/2018/844/oj.

[2] B. Dong, C. Cao, S.E. Lee, Applying support vector machines to predict building energy consumption in tropical region, Energy Build. 37 (5) (2005) 545–553, http://dx.doi.org/10.1016/j.enbuild.2004.09.009.

[3] H.-x. Zhao, F. Magoulès, A review on the prediction of building energy consumption, Renew. Sustain. Energy Rev. 16 (6) (2012) 3586–3592, http://dx.doi.org/10.1016/j.rser.2012.02.049.

[4] J. Kallio, J. Tervonen, P. Räsänen, R. Mäkynen, J. Koivusaari, J. Peltola, Forecasting office indoor CO2 concentration using machine learning with a one-year dataset, Build. Environ. 187 (2021) 107409, http://dx.doi.org/10.1016/j.buildenv.2020.107409.

[5] N. Fumo, A review on the basics of building energy estimation, Renew. Sustain. Energy Rev. 31 (2014) 53–60, http://dx.doi.org/10.1016/j.rser.2013.11.040.

[6] I. Khan, A. Capozzoli, S.P. Corgnati, T. Cerquitelli, Fault detection analysis of building energy consumption using data mining techniques, Energy Procedia 42 (2013) 557–566, http://dx.doi.org/10.1016/j.egypro.2013.11.057.

[7] X. Zhou, X. Yang, J. Ma, K.I.-K. Wang, Energy efficient smart routing based on link correlation mining for wireless edge computing in IoT, IEEE Internet Things J. (2021) 1, http://dx.doi.org/10.1109/JIOT.2021.3077937.

[8] I. ASHRAE, 2009 ASHRAE Handbook: Fundamentals, American Society of Heating, Refrigeration and Air-Conditioning Engineers, 2009, URL: http://www.ashrae.org.

[9] M.Y.L. Chew, K. Yan, Enhancing interpretability of data-driven fault detection and diagnosis methodology with maintainability rules in smart building management, J. Sensors 2022 (2022) http://dx.doi.org/10.1155/2022/5975816.

[10] F. Causone, S. Carlucci, M. Ferrando, A. Marchenko, S. Erba, A data-driven procedure to model occupancy and occupant-related electric load profiles in residential buildings for energy simulation, Energy Build. 202 (2019) 109342, http://dx.doi.org/10.1016/j.enbuild.2019.109342.

[11] R. Markovic, E. Grintal, D. Wölki, J. Frisch, C. van Treeck, Window opening model using deep learning methods, Build. Environ. 145 (2018) 319–329, http://dx.doi.org/10.1016/j.buildenv.2018.09.024.

[12] R. Markovic, Generic Occupant Behavior Modeling for Commercial Buildings (Ph.D. thesis), Doctoral Thesis. RWTH Aachen University, 2020, URL: http://publications.rwth-aachen.de/record/795601/files/795601.pdf.

[13] Y. Luo, Y. Zhang, X. Cai, X. Yuan, E^2GAN: 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, 2019, pp. 3094–3100, http://dx.doi.org/10.24963/ijcai.2019/429.

[14] A. Chong, K.P. Lam, W. Xu, O.T. Karaguzel, Y. Mo, Imputation of missing values in building sensor data, in: ASHRAE and IBPSA-USA SimBuild, Vol. 6, 2016, pp. 407–414, URL: https://www.researchgate.net/publication/306080123_IMPUTATION_OF_MISSING_VALUES_IN_BUILDING_SENSOR_DATA.

[15] L.M. Candanedo, V. Feldheim, D. Deramaix, Reconstruction of the indoor temperature dataset of a house using data driven models for performance evaluation, Build. Environ. 138 (2018) 250–261, http://dx.doi.org/10.1016/j.buildenv.2018.04.035.

[16] S.D. Sagan, Learning from normal accidents, Org. Environ. 17 (1) (2004) 15–19, http://dx.doi.org/10.1177/1086026603262029.

[17] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, Y. Li, BRITS: Bidirectional recurrent imputation for time series, in: S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 31, Curran Associates, Inc. 2018, URL: https://proceedings.neurips.cc/paper/2018/file/734e6bfcd358e25ac1db0a4241b95651-Paper.pdf.

[18] J. Yoon, W.R. Zame, M. van der Schaar, Estimating missing data in temporal data streams using multi-directional recurrent neural networks, IEEE Trans. Biomed. Eng. 66 (5) (2019) 1477–1490, http://dx.doi.org/10.1109/TBME.2018.2874712.

[19] J. Zhang, P. Yin, Multivariate time series missing data imputation using recurrent denoising autoencoder, in: 2019 IEEE International Conference on Bioinformatics and Biomedicine, BIBM, 2019, pp. 760–764, http://dx.doi.org/10.1109/BIBM47256.2019.8982996.

[20] Z. Guo, Y. Wan, H. Ye, A data imputation method for multivariate time series based on generative adversarial network, Neurocomputing 360 (2019) 185–197, http://dx.doi.org/10.1016/j.neucom.2019.06.007.

[21] F.M. Bianchi, L. Livi, K.Ø. Mikalsen, M. Kampffmeyer, R. Jenssen, Learning representations of multivariate time series with missing data, Pattern Recognit. 96 (2019) 106973, http://dx.doi.org/10.1016/j.patcog.2019.106973.

[22] I.R. White, P. Royston, A.M. Wood, Multiple imputation using chained equations: issues and guidance for practice, Stat. Med. 30 (4) (2011) 377–399, http://dx.doi.org/10.1002/sim.4067.

[23] A.T. Hudak, N.L. Crookston, J.S. Evans, D.E. Hall, M.J. Falkowski, Nearest neighbor imputation of species-level, plot-scale forest structure attributes from LiDAR data, Remote Sens. Environ. 112 (5) (2008) 2232–2245, http://dx.doi.org/10.1016/j.rse.2007.10.009.

[24] G.E. Box, G.M. Jenkins, G.C. Reinsel, G.M. Ljung, Time series analysis: forecasting and control, J. Oper. Res. Soc. 22 (2015) 199–201, http://dx.doi.org/10.1111/jtsa.12194.

[25] J.W. Galbraith, V. Zinde-Walsh, Autoregression-Based Estimators for ARFIMA Models, CIRANO Working Papers 2001s-11, CIRANO, 2001, URL: https://ideas.repec.org/a/rjr/romjef/vy2016i4p5-34.html.

[26] C. Hamzaçebi, Improving artificial neural networks' performance in seasonal time series forecasting, Inform. Sci. 178 (23) (2008) 4550–4559, http://dx.doi.org/10.1016/j.ins.2008.07.024.

[27] E. Acar, D.M. Dunlavy, T.G. Kolda, 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, 2010, pp. 701–712, http://dx.doi.org/10.1137/1.9781611972801.61.

[28] G.E. Batista, M.C. Monard, An analysis of four missing data treatment methods for supervised learning, Appl. Artif. Intell. 17 (5–6) (2003) 519–533, http://dx.doi.org/10.1080/713827181.

[29] S.L. Lauritzen, The EM algorithm for graphical association models with missing data, Comput. Statist. Data Anal. 19 (2) (1995) 191–201, http://dx.doi.org/10.1016/0167-9473(93)E0056-A.

[30] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. R. Stat. Soc. Ser. B Stat. Methodol. 39 (1) (1977) 1–22, http://dx.doi.org/10.1111/j.2517-6161.1977.tb01600.x.

[31] Z. Wang, L. Wang, Y. Tan, J. Yuan, Fault detection based on Bayesian network and missing data imputation for building energy systems, Appl. Therm. Eng. 182 (2021) 116051, http://dx.doi.org/10.1016/j.applthermaleng.2020.116051.

[32] L. Zhou, C. Ma, X. Shi, D. Zhang, W. Li, 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, http://dx.doi.org/10.1109/IJCNN52387.2021.9534419.

[33] K. Yan, Chiller fault detection and diagnosis with anomaly detective generative adversarial network, Build. Environ. (2021) 107982, http://dx.doi.org/10.1016/j.buildenv.2021.107982.

[34] K. Yan, A. Chong, Y. Mo, Generative adversarial network for fault detection diagnosis of chillers, Build. Environ. 172 (2020) 106698, http://dx.doi.org/10.1016/j.buildenv.2020.106698.

[35] X. Zhou, X. Xu, W. Liang, Z. Zeng, Z. Yan, Deep-learning-enhanced multitarget detection for end–edge–cloud surveillance in smart IoT, IEEE Internet Things J. 8 (16) (2021) 12588–12596, http://dx.doi.org/10.1109/JIOT.2021.3077449.

[36] X. Zhou, W. Liang, W. Li, K. Yan, S. Shimizu, K.I.-K. Wang, Hierarchical adversarial attacks against graph neural network based IoT network intrusion detection system, IEEE Internet Things J. (2021) 1, http://dx.doi.org/10.1109/JIOT.2021.3130434.

[37] C. Chen, K. Li, W. Wei, J.T. Zhou, Z. Zeng, Hierarchical graph neural networks for few-shot learning, IEEE Trans. Circuits Syst. Video Technol. 32 (1) (2022) 240–252, http://dx.doi.org/10.1109/TCSVT.2021.3058098.

[38] C. Chen, K. Li, X. Zou, Z. Cheng, W. Wei, Q. Tian, Z. Zeng, Hierarchical semantic graph reasoning for train component detection, IEEE Trans. Neural Netw. Learn. Syst. (2021) 1–13, http://dx.doi.org/10.1109/TNNLS.2021.3057792.

[39] Z. Che, S. Purushotham, K. Cho, D. Sontag, Y. Liu, Recurrent neural networks for multivariate time series with missing values, Sci. Rep. 8 (1) (2018) 6085, http://dx.doi.org/10.1038/s41598-018-24271-9.

[40] A. Rahman, V. Srikumar, A.D. Smith, Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks, Appl. Energy 212 (2018) 372–385, http://dx.doi.org/10.1016/j.apenergy.2017.12.051.

[41] J. Ma, J.C. Cheng, F. Jiang, W. Chen, M. Wang, C. Zhai, A bi-directional missing data imputation scheme based on LSTM and transfer learning for building energy data, Energy Build. 216 (2020) 109941, http://dx.doi.org/10.1016/j.enbuild.2020.109941.

[42] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems, Vol. 27, Curran Associates, Inc. 2014, URL: https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

[43] J. Yoon, J. Jordon, M. van der Schaar, GAIN: Missing data imputation using generative adversarial nets, in: J. Dy, A. Krause (Eds.), Proceedings of the 35th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 80, PMLR, 2018, pp. 5689–5698, URL: https://proceedings.mlr.press/v80/yoon18a.html.

[44] Y. Luo, X. Cai, Y. ZHANG, J. Xu, Y. xiaojie, Multivariate time series imputation with generative adversarial networks, in: S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 31, Curran Associates, Inc. 2018, URL: https://proceedings.neurips.cc/paper/2018/file/96b9bff013acedfb1d140579e2fbeb63-Paper.pdf.

[45] Y. Zhang, B. Zhou, X. Cai, W. Guo, X. Ding, X. Yuan, Missing value imputation in multivariate time series with end-to-end generative adversarial networks, Inform. Sci. 551 (2021) 67–82, http://dx.doi.org/10.1016/j.ins.2020.11.035.

[46] X. Lai, X. Wu, L. Zhang, W. Lu, C. Zhong, Imputations of missing values using a tracking-removed autoencoder trained with incomplete data, Neurocomputing 366 (2019) 54–65, http://dx.doi.org/10.1016/j.neucom.2019.07.066.

[47] J. Loy-Benitez, S. Heo, C. Yoo, Imputing missing indoor air quality data via variational convolutional autoencoders: Implications for ventilation management of subway metro systems, Build. Environ. 182 (2020) 107135, http://dx.doi.org/10.1016/j.buildenv.2020.107135.

[48] Y. Duan, Y. Lv, Y.-L. Liu, F.-Y. Wang, An efficient realization of deep learning for traffic data imputation, Transp. Res. C 72 (2016) 168–181, http://dx.doi.org/10.1016/j.trc.2016.09.015.

[49] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceedings of the 25th International Conference on Machine Learning, 2008, pp. 1096–1103, http://dx.doi.org/10.1145/1390156.1390294.

[50] W. Li, L. Xu, Z. Liang, S. Wang, J. Cao, C. Ma, X. Cui, Sketch-then-edit generative adversarial network, Knowl.-Based Syst. 203 (2020) 106102, http://dx.doi.org/10.1016/j.knosys.2020.106102.

[51] Z. Li, C. Ma, X. Shi, D. Zhang, W. Li, 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, http://dx.doi.org/10.1109/IJCNN52387.2021.9534001.

[52] W. Li, L. Fan, Z. Wang, C. Ma, X. Cui, Tackling mode collapse in multi-generator GANs with orthogonal vectors, Pattern Recognit. 110 (2021) 107646, http://dx.doi.org/10.1016/j.patcog.2020.107646.

[53] C. Chen, K. Li, S.G. Teo, X. Zou, K. Li, Z. Zeng, Citywide traffic flow prediction based on multiple gated spatio-temporal convolutional neural networks, ACM Trans. Knowl. Discov. Data 14 (4) (2020) http://dx.doi.org/10.1145/3385414.

[54] C. Ma, X. Shi, W. Zhu, W. Li, X. Cui, 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, http://dx.doi.org/10.1109/MSN48538.2019.00082.

[55] C. Ma, X. Shi, W. Li, W. Zhu, Edge4TSC: Binary distribution tree-enabled time series classification in edge environment, Sensors 20 (7) (2020) http://dx.doi.org/10.3390/s20071908.

[56] J. Liu, GAMS indoor air quality dataset, 2013, GitHub Repository. GitHub, https://github.com/twairball/gams-dataset.

[57] H. Zhang, R. Srinivasan, X. Yang, Simulation and analysis of indoor air quality in florida using time series regression (TSR) and artificial neural networks (ANN) models, Symmetry 13 (6) (2021) 952, http://dx.doi.org/10.3390/sym13060952.

[58] ANSI/ASHARE, ANSI/ASHRAE 62.1-2019: Ventilation for Indoor Air Quality, American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. 2019, URL: https://webstore.ansi.org/Standards/ASHRAE/ANSIASHRAE622019.

[59] H. Zhang, R. Srinivasan, V. Ganesan, Low cost, multi-pollutant sensing system using raspberry pi for indoor air quality monitoring, Sustainability 13 (1) (2021) 370, http://dx.doi.org/10.3390/su13010370.

[60] A. USEPA, A standardized EPA protocol for characterizing indoor air quality in large office buildings, 2016, Indoor Environment Division US EPA, Washington, DC.

[61] S. Bai, J.Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, 2018, CoRR, arXiv:1803.01271.

[62] A. Liguori, R. Markovic, T.T.H. Dam, J. Frisch, C. van Treeck, F. Causone, Indoor environment data time-series reconstruction using autoencoder neural networks, Build. Environ. 191 (2021) 107623, http://dx.doi.org/10.1016/j.buildenv.2021.107623.

[63] Eltonlaw, Impyute, 2017, GitHub Repository, GitHub, https://github.com/eltonlaw/impyute.

[64] L. van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (86) (2008) 2579–2605, URL: http://jmlr.org/papers/v9/vandermaaten08a.html.

[65] G.E. Hinton, S. Roweis, Stochastic neighbor embedding, in: S. Becker, S. Thrun, K. Obermayer (Eds.), Advances in Neural Information Processing Systems, Vol. 15, MIT Press, 2003, URL: https://proceedings.neurips.cc/paper/2002/file/6150ccc6069bea6b5716254057a194ef-Paper.pdf.

[66] K.Y. Wong, F.-l. Chung, Visualizing time series data with temporal matching based t-SNE, in: 2019 International Joint Conference on Neural Networks, IJCNN, 2019, pp. 1–8, http://dx.doi.org/10.1109/IJCNN.2019.8851847.

[67] K. Yang, C. Shahabi, A PCA-based similarity measure for multivariate time series, in: Proceedings of the 2nd ACM International Workshop on Multimedia Databases, 2004, pp. 65–74, http://dx.doi.org/10.1145/1032604.1032616.

[68] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Y.W. Teh, M. Titterington (Eds.), Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, in: Proceedings of Machine Learning Research, vol. 9, PMLR, Chia Laguna Resort, Sardinia, Italy, 2010, pp. 249–256, URL: https://proceedings.mlr.press/v9/glorot10a.html.

[69] R. Markovic, E. Azar, M.K. Annaqeeb, J. Frisch, C. van Treeck, Day-ahead prediction of plug-in loads using a long short-term memory neural network, Energy Build. 234 (2021) 110667, http://dx.doi.org/10.1016/j.enbuild.2020.110667.