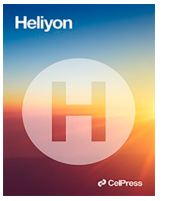
Heliyon 10 (2024) e24738

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon



Research article

A densely connected causal convolutional network separating past and future data for filling missing $PM_{2.5}$ time series data

Peng Yuan, Yiwen Jiao, Jiaxue Li, Yan Xia *

College of Computer Science and Electronic Engineering, Hunan University, Changsha, 410082, Hunan, China

ARTICLE INFO

Keywords:
Missing data filling
Air quality
Densely connected causal convolutional network
Deep learning
$PM_{2.5}$

ABSTRACT

Air pollution poses a significant threat to human health and the environment globally. Precise analysis and prediction of pollutant concentrations are essential for monitoring and managing air quality. However, reliable analysis and prediction require comprehensive and high-quality data, which is often compromised due to missing data during collection. Unfortunately, conventional methods for addressing missing data fall short of providing adequate solutions. The missing data for air quality indicators are commonly systematic, with all data points missing for extended periods. This makes it difficult to establish correlations and populate the missing data accurately. To address this problem, we propose a Densely Connected Causal Convolutional Network Separating Past and Future Data (DCCN-SPF), a deep learning-based model that fills in continuous missing $PM_{2.5}$ concentration data in the original dataset. It extracts features from past and future data separately using densely connected causal convolutional networks and incorporates linear interpolation and deep learning structures to improve prediction accuracy. Using air quality monitoring data from the China Environmental Monitoring Station between 2017 and 2021 in Beijing, we compare our proposed model with baseline models and find that our model outperforms others in predicting $PM_{2.5}$ concentrations. The evaluation metrics MAE and RMSE are used, revealing significant reductions of 8.7–21.6 % for MAE and 7.1–23.5 % for RMSE in favor of our proposed DCCN-SPF model.

1. Introduction

Air pollution is a major environmental issue that has ramifications for air quality, regional and global climate, and human health. According to a scientific statement by the World Health Organization (WHO), approximately 4.2 million people worldwide die each year due to exposure to ambient (outdoor) air pollution. Particulate pollutants, including $PM_{2.5}$ and PM_{10}, can be easily inhaled into the lungs due to their small size and can be very harmful to humans [1]. $PM_{2.5}$, in particular, poses a significant risk to human health and has been linked to a variety of ailments including stroke [2], asthma [3], and Alzheimer's disease [4] (AD) [5,6]. Therefore, the analysis and prediction of air pollutant concentrations, especially PM2.5 and PM_{10}, are considered essential issues for human health and environmental protection [7–10]. When conducting these studies, complete and high-quality air pollution data must be required to produce reliable results [11].

Conventional missing data processing methods are classified into two categories: direct deletion and traditional filling. The direct deletion method is less applicable to time-series data. This is because, for most time-series data prediction models, simply deleting the

* Corresponding author.
E-mail address: xiayan@hnu.edu.cn (Y. Xia).

https://doi.org/10.1016/j.heliyon.2024.e24738
Received 18 May 2023; Received in revised form 7 January 2024; Accepted 12 January 2024
Available online 17 January 2024

missing samples creates the illusion that there are no missing values, thus leading the model to incorrectly consider two non-continuous collected data as time-continuous (because the middle missing values are deleted). The model learns the incorrect data, which is not conducive to the prediction model discovering the essential features and development patterns implied by the time-series data. Cutting out data before and after events in order to prevent missing values may result in a reduction of the dataset, which would negatively impact the performance of the model. For time-series data, even a small number of missing values can result in a significant amount of data becoming unusable. The reduction in data volume can have a detrimental effect on the performance of the model [12, 13].

Traditional filling methods are based on statistical knowledge, such as linear interpolation methods like mean interpolation and median interpolation, which use the observed elements' corresponding mean or median to fill in the missing values. Regrettably, the data before and after the missing values are ignored, as is the time-series data's temporal correlation [14].

In recent years, more sophisticated methods for filling in missing time series data have been developed. For example, W.L. Junger et al. [15] applied an enhanced expectation maximization (EM) technique for missing data filling of air contaminants. Li et al. [16] proposed a least squares support vector machine, which is based on a multivariate time series filling approach (LSSVM). HASTIE et al. [17] employed a filling method based on matrix factorization (MF) for missing value filling. These approaches are difficult to model and entail sophisticated computing operations, hence they are utilized less regularly in practice.

With the emergence of deep learning, numerous neural network-based algorithms are being utilized to fill in missing data. For instance, Che et al. [18] developed a deep learning model for missing data interpolation based on gated recursive units (GRU). Eum et al. [15]established a deep learning model based on convolutional neural networks (CNN) and long and short-term models (LSTM) for long-term missing in personal time activity data. Ma et al. [11] proposed a new transferred LSTM-based iterative estimation model.

Unfortunately, whereas numerous techniques to estimate missing data have been proposed, the majority of them have been applied in the fields of clinical diseases, computer science, and economics [11], with only a few studies focusing on the problem of missing data in air pollution. The majority of current neural network-based missing time series data-filling methods are based on recurrent neural network (RNN) structures and their variants (e.g. LSTM, GRU). For RNNs, each time step is computed based on the results of the previous time step. Therefore, RNNs have limited computational parallelism, which slows down the training of RNN models [19].

The vast majority of methods are based on the assumption that missing data will occur at random, but statistical analysis has shown that it is more typical for all indicators to be absent for an extended period in air quality data, which is ignored by most methods currently in use. The approaches that rely on associations between variables to fill in the data. Several interpolation algorithms, such as MissForest [20] and MBGAN [21], have been proposed to fill in the missing data relying on correlations between variables. Unfortunately, statistical studies have shown that in some cases, such as power loss, equipment failure, network failure, or server failure, all measurements may be unavailable for an extended period until the issue is resolved [22]. In such cases, correlations between variables cannot be used to fill in the missing data. The air quality data exhibits significant time-series dependence, and missing values are correlated with the pre- and post-missing data. Instead of relying on correlations between variables, we suggest filling in the gaps with data from before and after the missing values.

A series of studies have demonstrated that neural networks exhibit outstanding performance in air quality prediction tasks. Oprea et al. successfully applied the method of combining artificial neural networks (ANN) and recurrent neural networks (RNN) for PM2.5 prediction [23]. Wang et al. combined genetic algorithms with artificial neural networks to forecast PM2.5 concentrations [24]. Huang et al. employed a gated recurrent unit (GRU) neural network based on empirical mode decomposition to predict short-term PM2.5 concentrations at a single station [25]. Chae et al. achieved excellent results in real-time prediction of PM10 and PM2.5 by utilizing an interpolation convolutional neural network [26]. Moreover, Yuan et al. combined multiple types of neural networks to further enhance the model's performance [27].

To solve the previously mentioned issue, in this paper, we propose DCCN-SPF for filling missing $PM_{2.5}$ time series data. We do not

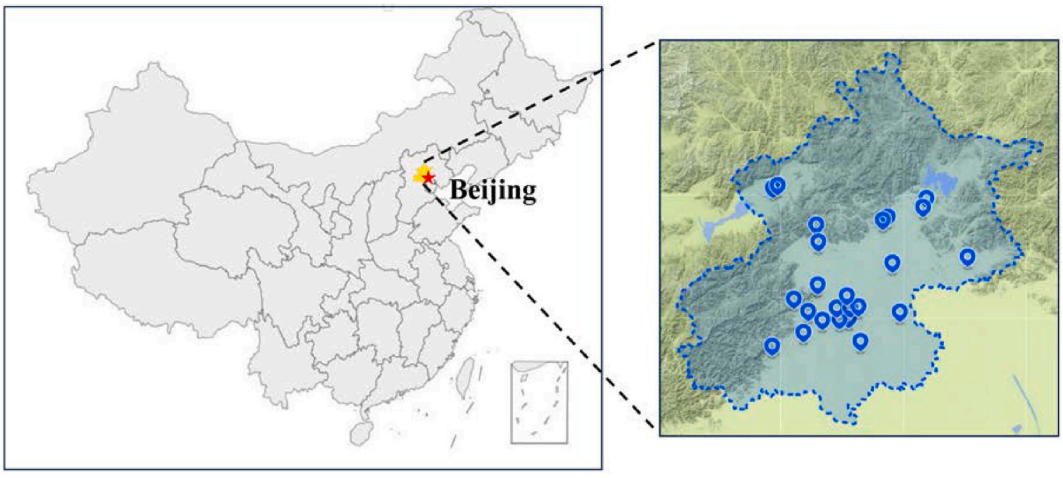


Fig. 1. Monitoring station map for air pollution in Beijing.

use the computationally limited RNN structure to extract data features. Rather, we employ the causal convolutional neural network, which significantly reduces training time to only 10 % of that of RNN models. We employ two feature extractors that process the data preceding and following the missing data independently and concurrently. This method of separating past and future data enables better data feature mining. We also use densely connected convolutional layers, which allows us to reuse features and avoid shallow features being lost in forward propagation.

We offer a simple and effective prior for the model: the missing date will be initially filled with the linear interpolation results based on the temporally nearest two existing values. Linear interpolation can provide a rough trend of missing data. In comparison to entirely using the neural network to predict missing data, incorporating prior knowledge into the model can enhance model performance.

2. Materials and methods

2.1. Dataset description

We obtained the air quality monitoring data of Beijing from 2017 to 2021 by running the crawler code. The original source of these data is the National Urban Air Quality Real-Time Release Platform of China Environmental Monitoring General Station (https://air.cnemc.cn:18007/). These monitoring data come from 24 monitoring points, and it is an average of the data from 24 monitoring points, reflecting the overall air quality of the city. Its geographical location is shown in Fig. 1. The latitude and longitude of these monitoring stations can be found in Table A1 in the appendix.

This dataset includes NO_2, SO_2, $PM_{2.5}$, PM_{10}, and other concentrations with 1-h as the minimum time scale as air quality indicators, and we selected $PM_{2.5}$ as the dataset for this study. In addition, we conducted an analysis of missing air quality data for 31 Chinese province capitals, as detailed in Fig. A1 in the appendix, which illustrates missing air quality data in the real world. We represent two types of missing data: random missing data for a single feature (Random Missing) and complete missing data for all data at the same time (Continuous Missing), which is defined in Fig. A2 in the appendix. And in Table A2 and Fig. A3, we calculate the number of missing air quality data for two categories in 31 cities between 2015 and 2021, and Table A3 shows the statistics on the length of time of continuous missing data and the corresponding number of occurrences.

The dataset in this study is divided into three parts: training set, validation set, and test set. To ensure the validity of the validation, we equalized the distribution of the three sets of data over time. We use the 2017–2020 data as the training set, the January–June 2021 data as the validation set, and the July–December 2021 data as the test set. We conducted all tests five times and averaged the results to assure the accuracy of the findings. The data was split into three sets based on time: training, validation, and testing. This approach emulates real-world scenarios and allows the model to learn from temporal patterns. The validation set aids in model selection and prevents overfitting, while the testing set evaluates the model's performance on unseen data. This method ensures a realistic assessment of the model's capabilities and avoids overfitting to future data.

2.2. Problem formulation

Regression is employed to fill in the missing $PM_{2.5}$ time series data. The data prior to and/or following the missing data are used as model inputs, and the forecasted value of the missing data is used as the model output.

We divide the time series of $PM_{2.5}$ concentrations into three components: $X_p = \{x_1, x_2, \ldots, x_{k1}\}$, $X_{l=} \{x_{k1+1}, x_{k1+2}, \ldots, x_{k1+n}\}$, $X_f = \{x_{k1+n+1}, x_{k1+n+2}, \ldots, x_{k1+n+k2}\}$, where X_p and X_f represent data before and after the missing data, with few or no missing values, respectively. X_l denotes missing data, where most or all of the data are missing. k_1 and k_2 denote time steps of past and future data, respectively, and n represents the duration of the missing data. Our objective is to discover a model M with inputs X_p and X_f and outputs Y, i.e. $Y = M([X_p, X_f])$. It is necessary for Y to fit closely to X_l.

2.3. Causal convolutional neural network

Convolutional neural networks (CNNs) have found broad applications in diverse fields. Specifically, 2D CNNs have been widely employed for image processing [28,29], whereas 3D CNNs have emerged as an active area of research for 3D imaging applications, including materials chemistry [30] and medical imaging [31]. Recent studies have demonstrated that 1D CNNs perform well in time series applications, such as urban water supply [32], fault detection [33], medical diagnosis [34]. In this study, we utilize a novel architecture based on the CNN structure to address the sequence modeling problem in the task of filling missing PM2.5 time series data.

The operation of CNN is based on multiplying each element in the convolution kernel with the corresponding element in a sub-region of the input data, and then summing the results to produce the feature map [35]. However, this mechanism is not appropriate for time series analysis since it fails to take into account the temporal dependency [35,36]. Causal convolution overcomes this restriction by maintaining the temporal sequence in which the data is modeled, thereby prohibiting the use of information from future data samples in analyzing any given time step [37].

Previous studies have shown that increasing the network depth can effectively enhance the model performance [36], leading to the incorporation of the multi-layer processing into the model. Although muti-layer model can expand the coverage of the historical data, it can only capture information from a linear timescale, making it unsuitable for gathering information over a lengthy period of time.

To access more historical data, we employed dilated convolutions following the works of Oord et al. and Bai et al [36]. In this 'dilation' scheme, a convolutional filter is applied over an area larger than its length by skipping input values at a given step, which

allows the receptive field of the network to be increased exponentially with respect to the number of dilated convolutional layers.

As the number of convolutional layers increases, the dilation factor becomes progressively larger. By choosing kernel sizes k and dilation factor d, the effective history of one such casual convolution layer is $(k-1)d$. The first layers are exposed to a smaller number of data samples and are expected to learn short-term patterns, while the latest layers can benefit from large receptive fields and can learn long-term trends. We provide an illustration in Fig. 2.

Based on Fig. 2 and the above description, the receptive field of the proposed causal convolution can be expanded in three ways:

(1) increasing the convolution kernel size k.
(2) increasing the expansion factor d.
(3) increasing the network depth l.

The number of levels (i.e. the depth of the network) and the kernel size are two important model parameters. The length of data that a neural network model can obtain is referred to as the model's receptive field, which is proportional to kernel size and the number of levels. We introduce d as the dilation factor, where $d = 2^i$ at level i of the network, l to be the number of levels, and k to be the kernel size. In this study, we exponentially increase d according to the network depth (i.e., $d = 2^i$ at level i of the network). The analysis of parameters is discussed in section 3.4.

2.4. Densely connected network

Convolutional neural networks (CNNs) are susceptible to gradient vanishing when they become too deep, which can lead to network degradation and failure of the model to converge. To address this issue, He et al. proposed the ResNet [38] architecture for training, which has been shown to effectively alleviate the problem of gradient vanishing in deep CNNs. Inspired by ResNet, Huang et al. found that creating short paths from early layers to later layers in the convolutional network can lead to more accurate and efficient training. They proposed the DenseNet [39] architecture, which introduces direct connections between any two layers with the same feature-map size (as shown in Fig. 3). Rather than mining data features from extremely deep or extensive architectures, DenseNet allows feature reuse throughout the networks to improve the network's characterization capabilities. Consequently, it can learn more compact and accurate models.

A dense block comprises multiple convolutional layers in series, with each layer taking the feature maps obtained by all previous layers as its input and feeding its feature map to all subsequent layers. This enhances feature propagation and promotes feature reuse, resulting in improved performance.

The dense block is composed of L convolutional layers. $X_1, X_2, \ldots, XL$, with the i-th layer receives the feature maps $X_0, X_1, \ldots, Xi\text{-}1$ of all the previous layers as input. The output X_i can be represented as Eq. (1).

$$X_i = H_i([X_{i-1}, X_{i-2}, \ldots\ldots, X_1, X_0]) \tag{1}$$

here, $Hi(\cdot)$ denotes a nonlinear composite function, which consists of two consecutive operations: convolution and ReLU. denotes concatenation, where the data is stitched together in the channel dimension.

For an input $X = \{x_0, x_1, \ldots, x_{n-1}\}$, the operation of convolution is as Eq. (2).

$$y = f\left(\sum\nolimits_i w_i x_i + b\right) \tag{2}$$

The activation function ReLu is as Eq. (3).

$$f(x) = \max(0, x)\begin{cases} x, if(x > 0) \\ 0, if(x \le 0) \end{cases} \tag{3}$$

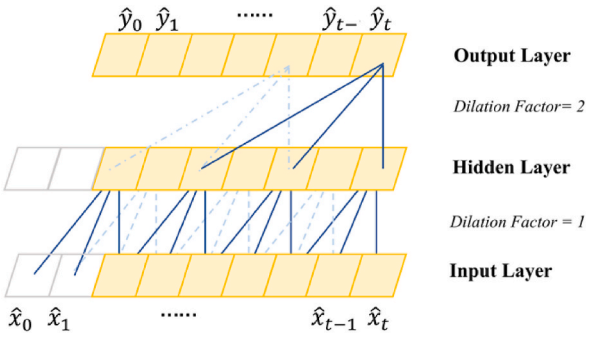


Fig. 2. A dilated causal convolution with dilation factor $d = 1, 2$ and kernel size $k = 3$.

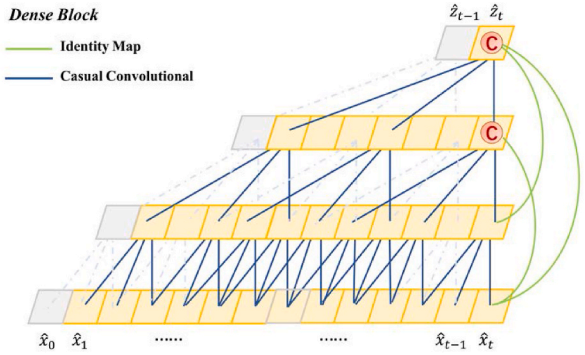


Fig. 3. An example of residual connection in dilated casual convolution network. The blue lines are kernels in the residual function, and the green lines are feature maps. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

Here w and b are the convolution kernel and offset respectively, and $f(\cdot)$ represents the activation function.

2.5. Densely connected causal convolutional network separating past and future data

In this section, we propose a new model, called DCCN-SPF, for filling missing $PM_{2.5}$ time series data. To process past and future data, the model employs two structurally identical feature extractors, each with a densely connected causal convolutional network. The outputs of the two feature extractors are stitched together and given to the linear layer for fitting. Adding the output of the linear layer and the linear interpolation, we obtain the predicted results.

Fig. 4 depicts the overall model's framework, which includes three major blocks: the Linear Interpolation unit, the Feature Extractor component and the Fit-and-Output unit.

(A) Linear Interpolation

We represented the original data as $X = [X_p, X_l, X_f]$, assuming that n consecutive time steps of data are missing, where the missing data is $X_l = \{x_{k1+1}, x_{k1+2}, \ldots, x_{k1+n}\}$. The previous k_1 time steps of X_l are the historical data set $X_p = \{x_1, x_2, \ldots, x_{k1}\}$, and the following k_2 time steps are the future data set $X_f = \{x_{k1+n+1}, x_{k1+n+2}, \ldots, x_{k1+n+k2}\}$.

The linear interpolation method uses x_{k1+1} and x_{k1+n+1} to calculate the general trend of the missing data. The results are denoted as S. Eq. (4), Eq. (5) and Eq. (6) show the specific calculation.

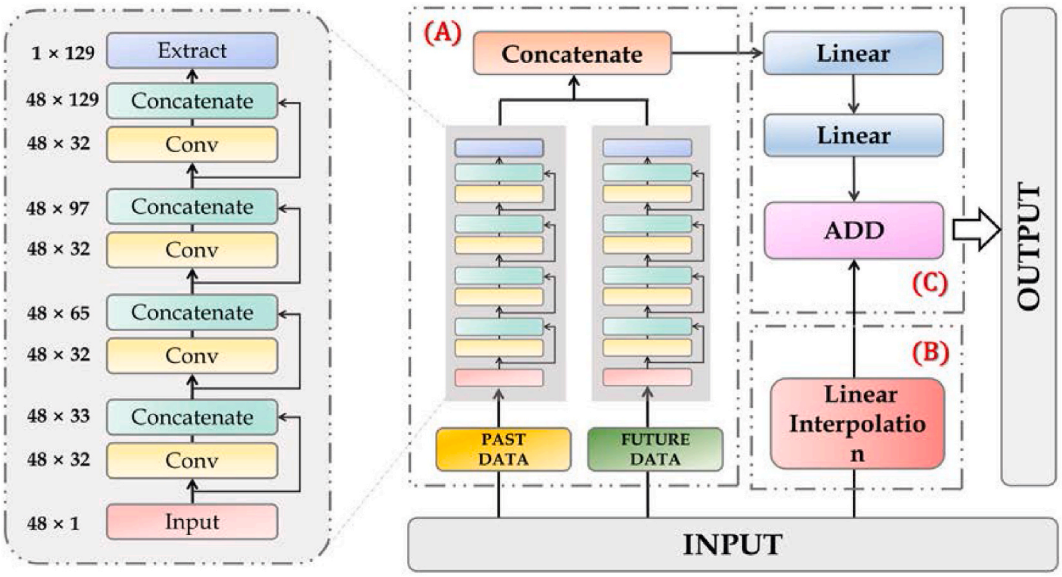


Fig. 4. The framework of the proposed DCCN-SPF model. The DCCN-SPF model can be divided into three components, (A) Linear Interpolation (B) Feature Extractor (C) Fit-and-Output.

$$\Delta = \frac{(x_{k1+n+1} - x_{k1})}{n+1} \tag{4}$$

$$s_i = x_{k1} + i \times \Delta \quad (1 \le i \le n) \tag{5}$$

$$S = [s_1, s_2, \ldots s_n] \tag{6}$$

(B) Feature Extractor

Fig. 5 shows the illustration of the convolutional network structure in the data feature extractor, using causal convolutional network with kernel size = 3 and dilations = 1, 2, and 4. The feature extractor is made up of multiple layers of densely connected causal convolutional layers. Each convolutional layer can reuse features from previous convolutional layers (including the input layer). The activation function for all convolutional layers is ReLU.

To extract features from historical data X_p and future data X_f we use two feature extractors. The causal convolution layer ensures that no information leaks from the past to the future when processing X_p. Note that when dealing with the future data, we take $X_f^{'}$ as the feature extractor input, which is the inverse order of X_f, i.e. $X_f^{'} = \{x_{k1+n+k2}, x_{k1+n+k2-1}, \ldots, x_{k1+n+1}\}$.

The output that the feature extractor produces is corresponding to the last dimension on the time axis (moment k_1 for historical data and moment $k_1 + n + 1$ for future data). The feature extractor output has m channels (m is related to the number of convolutional layers). Combining the output T_{past} (Eq. (7)) and T_{future} (Eq. (8)), we can obtain the ultimate prediction results T_{concat} (Eq. (9)).

$$T_{past} = \left[t_{k1}^{(1)}, \quad t_{k1}^{(2)}, \cdots\cdots \quad, t_{k1}^{(m)} \right] \tag{7}$$

$$T_{future} = \left[t_{k1+n+1}^{(1)}, \quad t_{k1+n+1}^{(2)}, \cdots\cdots \quad, t_{k1+n+1}^{(m)} \right] \tag{8}$$

$$T_{concat} = \left[t_{k1}^{(1)}, \quad t_{k1}^{(2)}, \cdots\cdots \quad, t_{k1}^{(m)}, \; t_{k1+n+1}^{(1)}, \quad t_{k1+n+1}^{(2)}, \cdots\cdots \quad, t_{k1+n+1}^{(m)} \right] \tag{9}$$

(C) Fit-and-Output

The module takes T_{concat} as input and passes it through 2 Linear Layers connected in series. The output dimension of the 2*nd* Linear Layer is the same number n of missing data, and the output feature vector $T = [t_1, t_2, \ldots t_n]$. The linear interpolation module's feature vector $S = [s_1, s_2, \ldots s_n]$ predicts the general trend of the missing data, and the data feature extractor's feature vector $T = [t_1, t_2, \ldots t_n]$ represents the deviation of the real data from S. By adding S and T, we can obtain the final predicted value for the missing $PM_{2.5}$ data, where the true value of the missing data is Y and the predicted value $\widehat{Y} = S + T$.

The mean square error (MSE) is chosen as the loss function for the training process, which is shown in Eq. (10). And the Adam optimizer is used to perform gradient descent optimization on the model's loss function.

$$LOSS = MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - \widehat{y_i})^2 \tag{10}$$

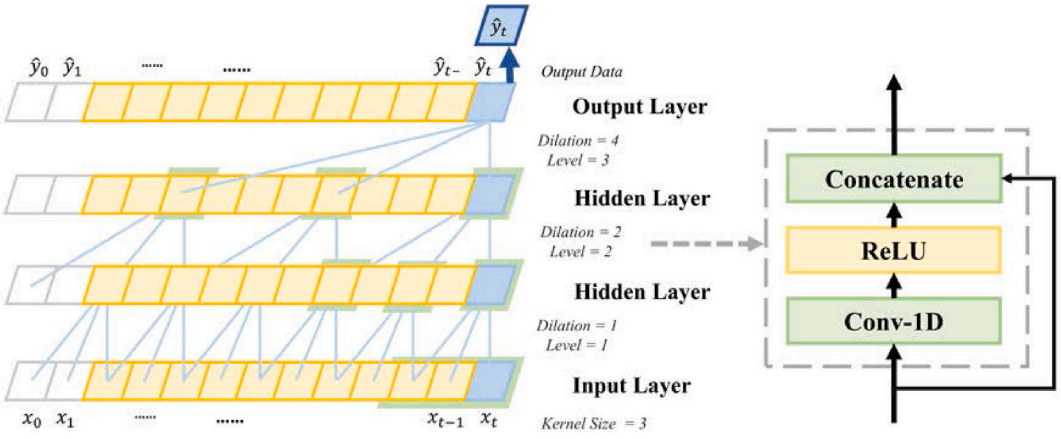


Fig. 5. Illustration of the convolutional network structure in the data feature extractor. Each layer applies Conv-1D, ReLU on the input in turn and concatenates with the feature maps from the previous layers.

2.6. Models construction

2.6.1. Comparing experiment among algorithms

To evaluate the effectiveness of the proposed DCCN-SPF, we compared its performance with other existing methods. Specifically, we considered classical statistical models and some modern deep-learning models, including a CNN proposed in Ref. [40] ANNs proposed in Ref. [41], CNN-LSTM proposed in Ref. [15], and a multi-head attention-based transformer model proposed in Ref. [42]. In addition, we also used the statistical method of linear interpolation and ML models such as K-nearest Neighbors (KNN) as baseline models for comparison.

2.6.2. Ablation experiment settings

In this study, we conduct ablation experiments on 3 aspects: the neural network layer connection method, whether to separate past and future data, and whether to incorporate the priory knowledge.

In the study of the neural network layer connection method, we choose three connection methods of neural networks to explore the effects of different connection methods on model performance, and the different compositions of X_i are demonstrated in Eq. (11), Eq. (12) and Eq. (13).

Densely connection (DCCN-SPF):

$$X_i = H_i([X_{i-1}, X_{i-2}, ..., X_1, X_0]) \tag{11}$$

Sequential connection:

$$X_i = H_i(X_{i-1}) \tag{12}$$

Residual connection:

$$X_i = H_i(X_{i-1}) + X_{i-1} \tag{13}$$

Since the RNNs model demands continuous input data, the presence of missing data leads to the truncation of historical and future data. Consequently, we attempt to use two independently trained feature extractors to extract features from historical and future data separately, merge them, and then input them into the linear layer to produce the ultimate prediction results. To investigate whether dividing the input data into historical and future data for independent processing affects the prediction results, we set up a control group DCCN, combined historical and future data for processing, and completed feature extraction using a feature extractor with the same structure as DCCN-SPF, and kept the remainder of the model framework consistent with DCCN-SPF. Eq. (14), Eq. (15) and Eq. (16) respectively show the form of input data for different models.

Input of DCCN-SPF (Separated):

$$X_{past} = \{x_1, x_2, \ldots, x_{k1}\} \tag{14}$$

$$X_{future} = \{x_{k1+n+k2}, x_{k1+n+k2-1}, \ldots, x_{k1+n+1}\} \tag{15}$$

Input of DCCN (Unseparated):

$$X = \begin{bmatrix} x_1, & x_2, \cdots\cdots, x_{k1} \\ x_{k1+n+k2}, & x_{k1+n+k2-1}, \cdots\cdots, x_{k1+n+1} \end{bmatrix} \tag{16}$$

In section 2.5, the ultimate output of linear layers is represented as $T = [t_1, t_2, ..., t_n]$ The output of the linear interpolation component is represented as $S = [s_1, s_2, ..., s_n]$. For DCCN-SPF, the output of the model is noted as $\widehat{Y}$, which is shown as Eq. (17):

$$\widehat{Y} = S + T \tag{17}$$

we compare DCCN-SPF with the following two models to investigate the optimal technique to use linear interpolation as a priori knowledge.

W/o interpolation: There is no usage of linear interpolation. The output of the model is as Eq. (18):

$$\widehat{Y} = T \tag{18}$$

weighted-sum: Introduce two trainable weight vectors W_1 (Eq. (19)) and W_2 (Eq. (20)). $\widehat{Y}$ is obtained by weighted summation of S and T as Eq. (21).

$$W_1 = [w_{11}, w_{12}, ..., w_{1n}] \tag{19}$$

$$W_2 = [w_{21}, w_{22}, ..., w_{2n}] \tag{20}$$

$$\widehat{Y} = W_1 \odot S + W_2 \odot T \tag{21}$$

3. Results and discussion

3.1. Experiment setup

All the experiments were executed on Intel(R) Core(TM) i7-12700 CPU devices with NVIDIA GeForce RTX 3090 Ti. The detailed training parameters were given in appendix B. Table B1, B2, B3, B4, B5, B6 demonstrate the training parameters of the DCCN-SPF, CNN, CNN-LSTM, Transformer, KNN and ANN model, respectively.

In section 3.2, we employed two performance metrics, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) to evaluate the performance of the model. For missing durations ranging from 1 h to 24 h, we compared the MAE and RMSE metrics for each missing duration to determine the prediction accuracy of the model.

MAE indicates the mean absolute difference between the true and predicted values and is insensitive to larger bias. RMSE indicates the standard deviation of the residuals and is more sensitive to larger error values, so it will provide greater penalties for larger deviations.

In section 3.3, We use the mean of MAE, RMSE and MAPE for each missing duration as the performance evaluation metric of the model. In section 3.4, We use the mean of MAE and RMSE as the performance evaluation metric of the model.

3.2. Model performance

First, we evaluated the performance of our model using MAE. Table 1 and Fig. 6 show the comparison between DCCN-SPF and Baseline models when using MAE to evaluate model performance. In summary, DCCN-SPF outperforms baseline models at all missing-time steps.

Generally, the MAE metrics of DCCN-SPF show a linear growth trend with increasing time step of missing data. In comparison, CNN-LSTM and Transformer's MAE metrics trends are relatively erratic and even exhibit regional non-monotonic fluctuation. These fluctuations illustrate that the CNN-LSTM and Transformer models have difficulties converging at times and cannot constantly guarantee strong prediction performance, whereas the DCCN-SPF model converges quickly and consistently.

When the missing time is limited, neural network methods (e.g., CNN and CNN-LSTM) outperform linear interpolation. Given its simplicity and effectiveness, linear interpolation is widely used.

We evaluated the performance of our model using RMSE. The RMSE of all models has an approximately linear increasing trend with increasing missing time. The comparison between DCCN-SPF and Baseline models when using RMSE is displayed in Table 2 and Fig. 7.

For all missing time steps, DCCN-SPF outperforms other baseline models. Besides, the disparity between DCCN-SPF and baseline models widens as the prediction horizon lengthens, suggesting that DCCN-SPF can increase padding accuracy.

Unlike the MAE metric, linear interpolation performs poorer than almost all neural network models on the RMSE metric. This is because RMSE is more sensitive to prediction values with huge biases. Linear interpolation may produce predictors with higher bias, resulting in enormous RMSEs.

In summary, DCCN-SPF substantially outperforms the other baseline models in both metrics MAE and RMSE on the task. We can observe that with the prediction horizon increasing, the difference between the predicted and true results will also increase. However,

Table 1
MAE performance of DCCN-SPF model and the baseline models when setting different periods at missing.

Time	Linear Interpolation	KNN	CNN	ANN	Transformer	CNN-LSTM	DCCN-SPF
1	0.64	0.86	0.61	0.69	0.81	0.79	**0.59**
2	0.96	1.11	0.85	0.96	1.06	1.03	**0.76**
3	1.29	1.40	1.09	1.28	1.36	1.27	**0.99**
4	1.64	1.81	1.39	1.60	1.68	1.51	**1.24**
5	1.98	2.09	1.67	1.91	2.00	2.03	**1.52**
6	2.31	2.45	1.97	2.24	2.42	2.09	**1.80**
7	2.63	2.70	2.24	2.58	2.62	2.44	**2.13**
8	2.94	3.02	2.52	2.94	2.99	2.79	**2.37**
9	3.23	3.33	2.81	3.21	3.18	3.08	**2.66**
10	3.52	3.62	3.12	3.56	3.56	3.26	**2.91**
11	3.80	3.95	3.37	3.89	3.74	3.63	**3.13**
12	4.08	4.23	3.62	4.15	3.98	4.06	**3.39**
13	4.35	4.50	3.95	4.42	4.47	4.19	**3.61**
14	4.60	4.77	4.31	4.78	4.52	4.45	**3.89**
15	4.84	5.01	4.53	5.14	4.98	4.90	**4.14**
16	5.07	5.24	4.82	5.23	5.20	5.37	**4.41**
17	5.28	5.51	5.00	5.53	5.41	5.24	**4.61**
18	5.48	5.67	5.20	5.75	5.44	5.37	**4.80**
19	5.67	5.91	5.52	5.98	5.80	5.74	**5.05**
20	5.85	6.11	5.64	6.15	6.25	6.01	**5.24**
21	6.02	6.37	5.92	6.29	6.13	6.20	**5.48**
22	6.18	6.55	6.17	6.67	6.63	6.46	**5.64**
23	6.32	6.76	6.33	6.71	7.10	6.83	**5.84**
24	6.45	6.93	6.51	6.88	6.94	7.09	**5.98**

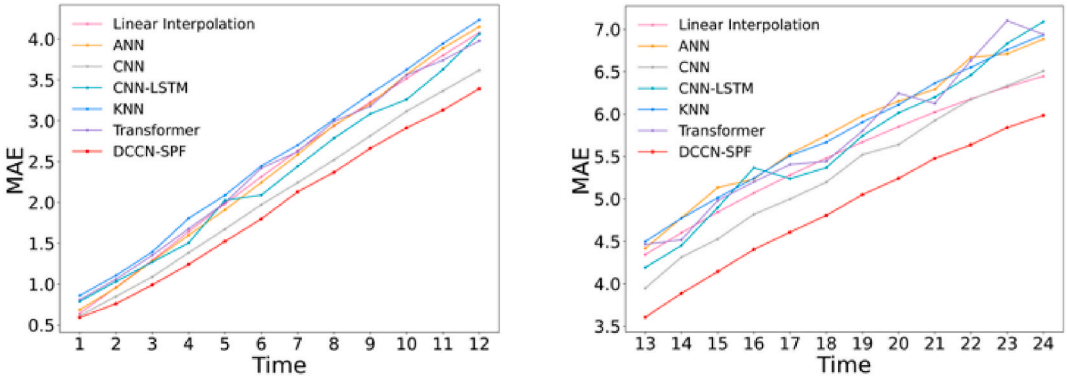


Fig. 6. MAE performance of DCCN-SPF model and the baseline models when setting different periods at missing.

Table 2
RMSE performance of DCCN-SPF model and the baseline models when setting different periods at missing.

Time	Linear Interpolation	KNN	CNN	ANN	Transformer	CNN-LSTM	DCCN-SPF
1	1.03	1.22	0.87	1.00	1.11	1.17	**0.84**
2	1.56	1.63	1.24	1.41	1.46	1.52	**1.12**
3	2.15	2.12	1.61	1.93	1.89	1.88	**1.49**
4	2.78	2.81	2.07	2.40	2.40	2.27	**1.90**
5	3.40	3.37	2.54	2.90	2.89	2.97	**2.38**
6	4.01	4.10	3.04	3.53	3.45	3.25	**2.85**
7	4.59	4.57	3.53	4.12	3.86	3.78	**3.46**
8	5.15	5.17	3.99	4.77	4.40	4.30	**3.92**
9	5.69	5.79	4.48	5.14	4.87	4.94	**4.36**
10	6.22	6.34	5.12	5.84	5.41	5.27	**4.77**
11	6.75	6.96	5.61	6.42	5.81	5.87	**5.16**
12	7.28	7.45	6.13	6.94	6.28	6.84	**5.62**
13	7.80	7.99	6.63	7.37	6.97	6.89	**6.07**
14	8.30	8.50	7.41	8.00	7.27	7.31	**6.71**
15	8.77	8.99	7.81	8.65	7.99	8.10	**7.23**
16	9.22	9.42	8.39	9.00	8.49	8.71	**7.69**
17	9.63	9.86	8.77	9.57	9.04	8.95	**8.09**
18	10.01	10.04	9.21	10.05	9.33	9.26	**8.52**
19	10.37	10.48	9.67	10.34	9.73	9.86	**8.92**
20	10.70	10.85	9.85	10.56	10.34	10.37	**9.33**
21	11.02	11.21	10.19	10.89	10.51	10.64	**9.64**
22	11.32	11.50	10.68	11.52	10.96	10.97	**10.06**
23	11.59	11.83	11.12	11.59	11.50	11.36	**10.33**
24	11.85	12.12	11.25	11.94	11.56	11.61	**10.65**

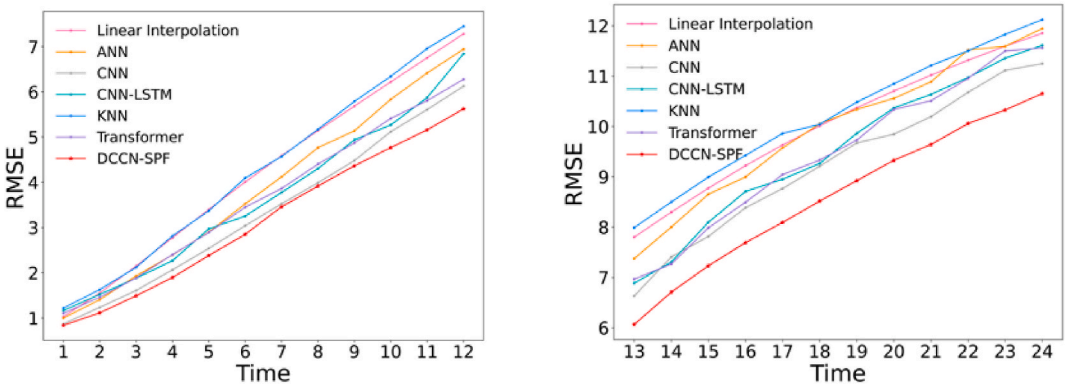


Fig. 7. RMSE performance of DCCN-SPF model and the baseline models when setting different periods at missing.

the prediction curves of our proposed DCCN-SPF model always match the real result best.

DCCN-SPF can both decrease the mean error between the prediction data and the true data and fill the extreme missing data efficiently, which makes it suitable for filling time series data with continuous missing values.

We suggest that linear interpolation is a good option when the length of missing data is limited and there is no strict requirement for filling accuracy. When high-precision filling or filling of extreme values is essential, DCCN-SPF is the preferable selection.

3.3. Ablation experiment results

3.3.1. Neural network layer connection method

We construct neural network models using different connection methods of neural network layers and compare the performance differences of the models with different connection methods. The results of the different models are shown in Table 3:

For the MAPE, MAE, and RMSE metrics, smaller values indicate better performance of the model. DCCN-SPF outperforms sequential connections in all evaluation metrics. The residual connection improves only in the MAPE metric and is less effective than the sequential connection in MAE and RMSE metrics. We can conclude that the dense connection method helps in data interpolation because it preserves the features extracted by the shallow network, which can then be reused by the deep network.

3.3.2. Input data processing method

As we expected, the DCCN-SPF that extracts features from historical and future data separately achieves better results. Compared with DCCN, our proposed DCCN-SPF reduces MAPE from 28.60 % to 25.23 %, a reduction of 3.37 %. MAE is reduced from 3.90 to 3.42, a reduction of 0.48, or about 12.3 %. RMSE is reduced from 6.33 to 5.88, a reduction of 0.45, or about 7.1 %. This illustrates that isolating historical and future data conforms to the actual law and helps to increase prediction accuracy.

3.3.3. Incorporate the prior knowledge

In accordance with section 2.6.2, we compared the performance of three models: w/o interpolation, weighted-sum, and DCCN-SPF. As shown in Table 4, The experimental results demonstrate that the utilization of linear interpolation in the proposed DCCN-SPF model effectively improves the prediction accuracy of the model.

The solution space of the weighted sum includes the solution space of w/o interpolation and DCCN-SPF. When $W_1 = 1$, $W_2 = 0$, the weighted-sum calculates in the same way as w/o interpolation. When $W_1 = 1$, $W_2 = 1$, the weighted-sum calculates in the same way as DCCN-SPF.

Intuitively, DCCN-SPF divides data prediction into two parts: the output of linear interpolation reflects the overall trend of missing data, and the neural network predicts the detailed fluctuations under the overall trend. It is more accurate to use the neural network to anticipate the fluctuation deviation rather than the entire missing data.

The weighted-sum model similarly employs a linear interpolation module, but its performance is highly reliant on the parameters W_1 and W_2. When the model selecting inappropriate weights, the prior knowledge expressed by the linear interpolation is not explicit and cannot be fully utilized.

3.4. Parameter influence on model performance

In DCCN-SPF, the feature extractor made up of the causal convolutional neural network is crucial for filling the missing $PM_{2.5}$ data. The convolutional kernel size k and the feature extractor's depth (the number of levels) l are two crucial variables.

Fig. 8 utilizes a heat map to demonstrate the difference in model performance for various kernel sizes and level combinations. We employ a grid search strategy, and the performance evaluation metrics are RMSE and MAE.

As shown in Fig. 8, the model achieves optima performance when the convolutional kernel size is 2 and the number of model layers is 4, resulting in an RMSE of 5.89 and an MAE of 3.43. Additionally, each convolution kernel has a corresponding number of layers to optimize the model.

When the convolution kernel is less than 4, the smaller the convolution kernel size, the more layers are necessary for the model to produce optimal outcomes. If the convolutional layer size is up to 4, the model only requires 2 layers to achieve the optimal performance.

The analysis of parameters d, k, and l is discussed in section 2.3. Using $R(l,k)$ to denote the receptive field of the l-th layer model with the kernel size k:

$$R(l,k) = d(l-1) \times (k-1) + R(l-1,k) \quad (l \geq 2) \tag{22}$$

Table 3
Comparison between DCCN-SPF, Sequential Connection and Residual Connection models. The results are the mean metrics of each training session.

Model	MAPE %	MAE	RMSE
Sequential Connection	26.04	3.48	5.93
Residual Connection	**24.29**	3.53	6.13
DCCN-SPF	25.23	**3.42**	**5.88**

Table 4
Comparison between DCCN-SPF, w/o interpolation and weighted-sum models. The results are the mean metrics of each training session.

Model	MAPE/%	MAE	RMSE
w/o interpolation	25.74	3.47	5.91
weighted-sum	25.76	3.47	5.94
DCCN-SPF	**25.23**	**3.42**	**5.88**

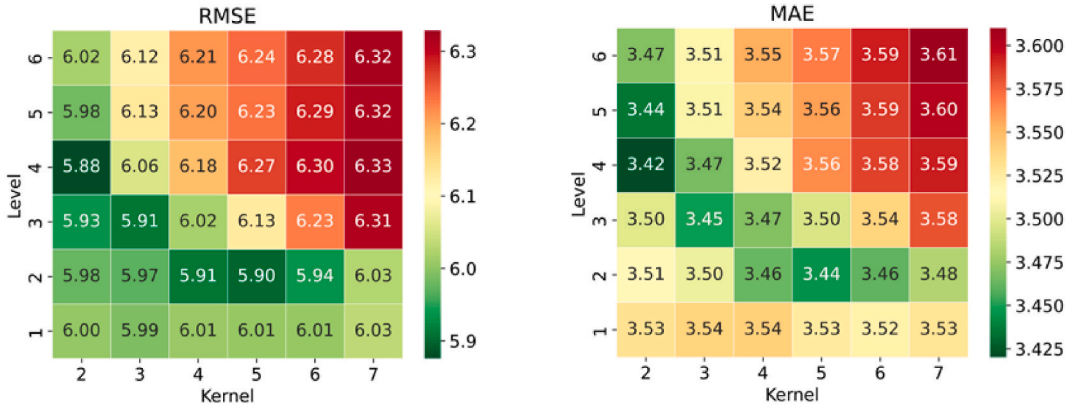


Fig. 8. Heat maps of model's MAE and RMSE metric performance. Data in the i-th row and j-th column represents the metrics under the combination of $(level, kernel\ size)$ is (i, j), for i ranging from 1 to 6 and j ranging from 2 to 7.

$$R(l,k) = k \quad (l = 1) \tag{23}$$

From Eq. (22) and Eq. (23), we can **obtain** Eq. (24)

$$R(l,k) = 2^l(k-1) - (k-2) \tag{24}$$

From Fig. 9, we can discover that the model performs best with a receptive field of 10–16. The impact of Level and Kernel size on ′ performance can be attributed to the following factors:

(1) When Level and Kernel are limited, the model can only rely on information from a relatively short period for data feature extraction and cannot collect enough information to fill in the missing data. This makes it difficult for the model to improve accuracy.
(2) When Level and Kernel are oversized, the model uses outdated information with no or only weak correlation to the missing values. Using this data does not provide sufficiently valid information for the data-filling task, but instead introduces noise, resulting in a decrease in prediction accuracy.

In practice, we recommend performing a grid search within a certain range of kernel size and dilation factor to determine the appropriate data duration. For instance, a model can only handle a maximum of 16 h of data, setting the input time to 48 h will result in many useless computations. By adjusting the data entry time from 48 h to 16 h, many invalid calculations can be avoided.

4. Conclusion

In our study, we observed that many datasets exhibit continuous missing values where all features are simultaneously absent. As a result, it becomes impossible to fill in these gaps using correlation between variables. To address this challenge, we utilized temporal relationships within the data to fill in the missing values instead of relying on correlations.

We propose the DCCN-SPF model. Unlike traditional RNN structures used to handle temporal dependencies, we leverage causal convolutional neural networks, which simplify the network architecture and can be computed in parallel. We input historical and future data into two independently trained feature extractors for parallel feature extraction and then use a linear layer for feature fitting. We investigated the effect of connection methods of causal convolutional networks, and the results demonstrate that dense connection is the superior connection method.

DCCN-SPF innovatively integrates advanced deep learning networks and linear interpolation to address the missing value problem. It incorporates prior knowledge into the model to reflect the general trend of the missing data. Our approach significantly enhances model performance by extracting historical and future data separately, optimizing further with densely connected networks and prior knowledge.

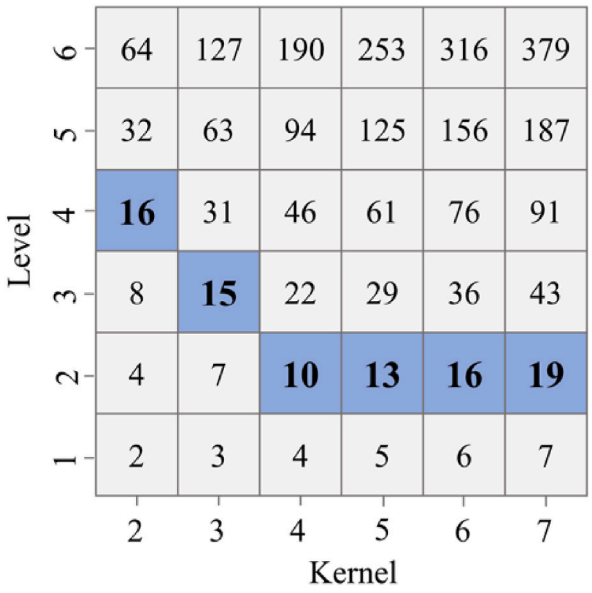


Fig. 9. Receptive field $R(l,k)$ for different combinations of kernel size and level, for k ranging from 1 to 6 and l ranging from 2 to 7.

Our proposed approach offers a novel solution for simplifying the structure of neural networks. We employ a data-driven methodology to calculate a suitable data length based on the number of levels and kernel size, enabling us to eliminate non-essential network structures and avoid unnecessary computations. This approach represents an innovative strategy for neural network architectures, providing a streamlined and effective framework for feature extraction and model training.

We evaluate model performance using MAE and RMSE. Results have shown that DCCN-SPF outperformed others algorithms with lower MAE and RMSE values. MAE was reduced by 8.7–21.6 % and RMSE was reduced by 7.1–23.5 %.

As figures shown in Table A3 in the appendix, most consecutive missing data in China's air quality dataset is within 16 h. Therefore, we set the maximum time limit for missing data to 16 h in our experiments. If we want to apply our model to other domains, we may need to select a suitable time threshold and consider increasing the time limit. Here are several approaches to increase the time limit:

1. More data: Generally, larger datasets improve the performance of the model.
2. Data augmentation: Apply augmentation methods to existing data to generate more diverse samples, although it may be more challenging for time series data.
3. Increase the time steps of input data: Longer time steps enable the model to capture more information. Adjust the model's layer structure and convolutional kernel size to fully utilize all input data.
4. Model transfer: Train the model based on pre-trained models to facilitate the development of a suitable model.

CRediT authorship contribution statement

Peng Yuan: Writing – review & editing, Visualization, Validation, Software, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Yiwen Jiao:** Writing – review & editing, Writing – original draft, Visualization, Validation, Investigation, Formal analysis. **Jiaxue Li:** Visualization, Validation, Data curation. **Yan Xia:** Writing – review & editing, Supervision, Software, Resources, Project administration, Methodology, Funding acquisition, Formal analysis.

Declaration of competing interest

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

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.heliyon.2024.e24738.

References

[1] A. Pozzer, S. Bacer, S.D.Z. Sappadina, F. Predicatori, A. Caleffi, Long-term concentrations of fine particulate matter and impact on human health in Verona, Italy, Atmospheric Pollution Research 10 (3) (May 2019) 731–738, https://doi.org/10.1016/j.apr.2018.11.012.

[2] Z. Chen, P. Liu, X. Xia, L. Wang, X. Li, The underlying mechanism of PM2.5-induced ischemic stroke, Environ. Pollut. 310 (Oct. 2022) 119827, https://doi.org/10.1016/j.envpol.2022.119827.

[3] C.A. Paterson, R.A. Sharpe, T. Taylor, K. Morrissey, Indoor PM2.5, VOCs and asthma outcomes: a systematic review in adults and their home environments, Environ. Res. 202 (Nov. 2021) 111631, https://doi.org/10.1016/j.envres.2021.111631.

[4] K. Thiankhaw, N. Chattipakorn, S.C. Chattipakorn, PM2.5 exposure in association with AD-related neuropathology and cognitive outcomes, Environ. Pollut. 292 (Jan. 2022) 118320, https://doi.org/10.1016/j.envpol.2021.118320.

[5] F. Li, et al., PM2.5-bound heavy metals from the major cities in China: spatiotemporal distribution, fuzzy exposure assessment and health risk management, J. Clean. Prod. 286 (Mar. 2021) 124967, https://doi.org/10.1016/j.jclepro.2020.124967.

[6] N.R. Martins, G. Carrilho da Graça, Impact of PM2.5 in indoor urban environments: a review, Sustain. Cities Soc. 42 (Oct. 2018) 259–275, https://doi.org/10.1016/j.scs.2018.07.011.

[7] P. Perez, C. Menares, C. Ramírez, PM2.5 forecasting in Coyhaique, the most polluted city in the Americas, Urban Clim. 32 (Jun. 2020) 100608, https://doi.org/10.1016/j.uclim.2020.100608.

[8] A. Mukundan, C.-C. Huang, T.-C. Men, F.-C. Lin, H.-C. Wang, Air pollution detection using a novel snap-shot hyperspectral imaging technique, Sensors 22 (16) (Jan. 2022), https://doi.org/10.3390/s22166231. Art. no. 16.

[9] C.-W. Chen, Y.-S. Tseng, A. Mukundan, H.-C. Wang, Air pollution: sensitive detection of PM2.5 and PM10 concentration using hyperspectral imaging, Appl. Sci. 11 (10) (Jan. 2021), https://doi.org/10.3390/app11104543. Art. no. 10.

[10] A. Mukundan, N. Hong-Thai, H.-C. Wang, Detection of PM 2.5 particulates using a snap-shot hyperspectral imaging technology. Proceedings of the 2022 Conference on Lasers and Electro-Optics Pacific Rim, CPDP_08, 2022, https://doi.org/10.1364/CLEOPR.2022.CPDP_08.

[11] J. Ma, et al., Transfer learning for long-interval consecutive missing values imputation without external features in air pollution time series, Adv. Eng. Inf. 44 (Apr. 2020) 101092, https://doi.org/10.1016/j.aei.2020.101092.

[12] J. Kaiser, Dealing with Missing Values in Data, JoSI, 2014, pp. 42–51, https://doi.org/10.20470/jsi.v5i1.178.

[13] L. Silva, L. Zárate, A brief review of the main approaches for treatment of missing data, Intell. Data Anal. 18 (Oct. 2014) 1177–1198, https://doi.org/10.3233/IDA-140690.

[14] Y. Luo, X. Cai, Y. Zhang, and J. Xu, "Multivariate Time Series Imputation with Generative Adversarial Networks".

[15] Y. Eum, E.-H. Yoo, Imputation of missing time-activity data with long-term gaps: a multi-scale residual CNN-LSTM network model, Comput. Environ. Urban Syst. 95 (Jul. 2022) 101823, https://doi.org/10.1016/j.compenvurbsys.2022.101823.

[16] Z.-X. Li, S.-H. Wu, C. Li, Y. Zhang, Research on methods of filling missing data for multivariate time series, in: 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA), Mar. 2017, pp. 382–385, https://doi.org/10.1109/ICBDA.2017.8078845.

[17] T. Hastie, R. Mazumder, J. Lee, R. Zadeh, Matrix Completion and Low-Rank SVD via Fast Alternating Least Squares, 2014, https://doi.org/10.48550/arXiv.1410.2596 arXiv, Oct. 09.

[18] Z. Che, S. Purushotham, K. Cho, D. Sontag, Y. Liu, Recurrent neural networks for multivariate time series with missing values, Sci. Rep. 8 (1) (Apr. 2018), https://doi.org/10.1038/s41598-018-24271-9. Art. no. 1.

[19] J. Bradbury, S. Merity, C. Xiong, R. Socher, Quasi-Recurrent Neural Networks, 2016, https://doi.org/10.48550/arXiv.1611.01576 arXiv, Nov. 21.

[20] D.J. Stekhoven, P. Bühlmann, MissForest–non-parametric missing value imputation for mixed-type data, Bioinformatics 28 (1) (Jan. 2012) 112–118, https://doi.org/10.1093/bioinformatics/btr597.

[21] Q. Ni, X. Cao, MBGAN: an improved generative adversarial network with multi-head self-attention and bidirectional RNN for time series imputation, Eng. Appl. Artif. Intell. 115 (Oct. 2022) 105232, https://doi.org/10.1016/j.engappai.2022.105232.

[22] X. Chen, J. Yang, L. Sun, A nonconvex low-rank tensor completion model for spatiotemporal traffic data imputation, Transport. Res. C Emerg. Technol. 117 (Aug. 2020) 102673, https://doi.org/10.1016/j.trc.2020.102673.

[23] M. Oprea, M. Popescu, S.F. Mihalache, A neural network based model for PM2.5 air pollutant forecasting, in: 2016 20th International Conference on System Theory, Control and Computing (ICSTCC), Oct. 2016, pp. 776–781, https://doi.org/10.1109/ICSTCC.2016.7790762.

[24] X. Wang, B. Wang, Research on prediction of environmental aerosol and PM2.5 based on artificial neural network, Neural Comput. Appl. 31 (12) (Dec. 2019) 8217–8227, https://doi.org/10.1007/s00521-018-3861-y.

[25] G. Huang, X. Li, B. Zhang, J. Ren, PM2.5 concentration forecasting at surface monitoring sites using GRU neural network based on empirical mode decomposition, Sci. Total Environ. 768 (May 2021) 144516, https://doi.org/10.1016/j.scitotenv.2020.144516.

[26] S. Chae, J. Shin, S. Kwon, S. Lee, S. Kang, D. Lee, PM10 and PM2.5 real-time prediction models using an interpolated convolutional neural network, Sci. Rep. 11 (1) (Jun. 2021) 11952, https://doi.org/10.1038/s41598-021-91253-9.

[27] P. Yuan, Y. Mei, Y. Zhong, Y. Xia, L. Fang, A hybrid deep learning model for predicting PM2.5, in: 2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP), Apr. 2022, pp. 274–278, https://doi.org/10.1109/ICSP54964.2022.9778520.

[28] D. Baur, K. Kroboth, C.-E. Heyde, A. Voelker, Convolutional neural networks in spinal magnetic resonance imaging: a systematic review, World Neurosurg. 166 (Oct. 2022) 60–70, https://doi.org/10.1016/j.wneu.2022.07.041.

[29] C.-L. Tsai, et al., Hyperspectral imaging combined with artificial intelligence in the early detection of esophageal cancer, Cancers 13 (18) (Jan. 2021), https://doi.org/10.3390/cancers13184593. Art. no. 18.

[30] T. Cawte, A. Bazylak, A 3D convolutional neural network accurately predicts the permeability of gas diffusion layer materials directly from image data, Curr. Opin. Electrochem. 35 (Oct. 2022) 101101, https://doi.org/10.1016/j.coelec.2022.101101.

[31] S. Niyas, S.J. Pawan, M. Anand Kumar, J. Rajan, Medical image segmentation with 3D convolutional neural networks: a survey, Neurocomputing 493 (Jul. 2022) 397–413, https://doi.org/10.1016/j.neucom.2022.04.065.

[32] L. Chen, et al., Short-term water demand forecast based on automatic feature extraction by one-dimensional convolution, J. Hydrol. 606 (Mar. 2022) 127440, https://doi.org/10.1016/j.jhydrol.2022.127440.

[33] R.F.R. Junior, I.A. dos, S. Areias, M.M. Campos, C.E. Teixeira, L.E.B. da Silva, G.F. Gomes, Fault detection and diagnosis in electric motors using 1d convolutional neural networks with multi-channel vibration signals, Measurement 190 (Feb. 2022) 110759, https://doi.org/10.1016/j.measurement.2022.110759.

[34] E.S. Ho, Z. Ding, Electrocardiogram analysis of post-stroke elderly people using one-dimensional convolutional neural network model with gradient-weighted class activation mapping, Artif. Intell. Med. 130 (Aug. 2022) 102342, https://doi.org/10.1016/j.artmed.2022.102342.

[35] S. Bai, J.Z. Kolter, V. Koltun, An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling, 2018, https://doi.org/10.48550/arXiv.1803.01271 arXiv, Apr. 19.

[36] L. Zhang, J. Na, J. Zhu, Z. Shi, C. Zou, L. Yang, Spatiotemporal causal convolutional network for forecasting hourly PM2.5 concentrations in Beijing, China, Comput. Geosci. 155 (Oct. 2021) 104869, https://doi.org/10.1016/j.cageo.2021.104869.

[37] S. Mariani, Q. Rendu, M. Urbani, C. Sbarufatti, Causal dilated convolutional neural networks for automatic inspection of ultrasonic signals in non-destructive evaluation and structural health monitoring, Mech. Syst. Signal Process. 157 (Aug. 2021) 107748, https://doi.org/10.1016/j.ymssp.2021.107748.

[38] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, 2015, https://doi.org/10.48550/arXiv.1512.03385 arXiv, Dec. 10.

[39] G. Huang, Z. Liu, L. van der Maaten, K.Q. Weinberger, Densely Connected Convolutional Networks, 2018, https://doi.org/10.48550/arXiv.1608.06993 arXiv, Jan. 28.

[40] J. B. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep Convolutional Neural Networks on Multichannel Time Series for Human Activity Recognition".
[41] S. Moghanlo, et al., Using artificial neural networks to model the impacts of climate change on dust phenomenon in the Zanjan region, north-west Iran, Urban Clim. 35 (Jan. 2021) 100750, https://doi.org/10.1016/j.uclim.2020.100750.
[42] S. Reza, M.C. Ferreira, J.J.M. Machado, J.M.R.S. Tavares, A multi-head attention-based transformer model for traffic flow forecasting with a comparative analysis to recurrent neural networks, Expert Syst. Appl. 202 (Sep. 2022) 117275, https://doi.org/10.1016/j.eswa.2022.117275.