C^2-Guard: A Cross-Correlation Gaining Framework for Urban Air Quality Prediction

Yu Chu[1], Lin Li[1] (✉), Qing Xie[1], and Guandong Xu[2]

[1] Wuhan University Of Technology, China
{chory7437,cathylilin,felixxq}@whut.edu.cn
[2] University of Technology Sydney, Australia guandong.xu@uts.edu.au

Abstract. Predicting air quality is increasingly important for protecting people's daily health and helping government decision-making. The multistep air quality prediction largely depends on the correlations of air quality-related factors. How to model the correlations among factors is a big challenge. In this paper, we propose a cross-correlation gaining framework (C^2-Guard) consisting of a temporal correlation module, factor correlation module, and cross gaining module for air quality (mainly PM2.5) prediction. Specifically, the temporal correlation module is used to extract the temporal dependence of air pollutant time series to gain their distributed representation. In the factor correlation module, a novel convolution and recalibration block is designed for air quality factor correlations extraction to gain their distributed representation in the factor dimension. In the cross gaining module, a joint-representation block is proposed to learn the cross-correlations between time and factor dimensions. Finally, extensive experiments are conducted on two real-world air quality datasets. The results demonstrate that our C^2-Guard outperforms the state-of-the-art methods of air pollutants prediction in terms of RMSE and MAE.

Keywords: Air quality prediction · Temporal correlation · Factor correlation · Cross correlation learning

1 Introduction

With the development of industrialization and urbanization, the air pollution problem has become increasingly serious. According to the Health Effects Institute (HEI), air pollution (PM2.5, ozone, and household air pollution) is the fifth leading risk factor for mortality worldwide. In 2017, air pollution is estimated to have contributed to close to 5 million deaths globally — nearly 1 in every 10 deaths[3]. Therefore, predicting changes in air pollutants that seriously affect urban air quality is of great significance for protecting people's daily health and helping government decision-making.

[3] https://www.healtheffects.org/announcements/state-global-air-2019-air-pollution-significant-risk-factor-worldwide

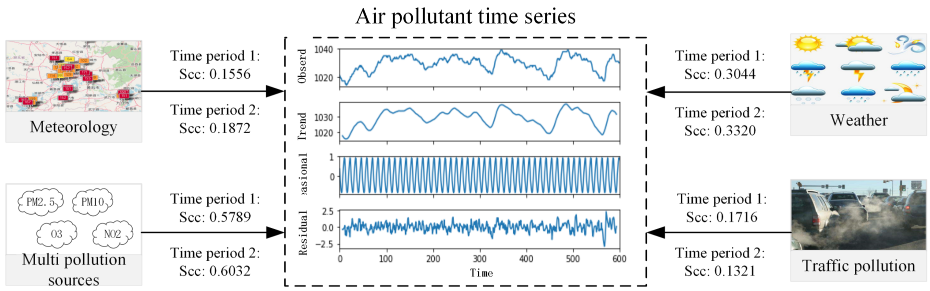


Fig. 1. The correlation analysis of air pollution time series. The center part is STL decomposition of air pollution time series. The two sides are Spearman correlation coefficients(Scc) of multiple air quality influential factors in different time periods.

In recent years, great efforts have been made on air quality prediction. First, air quality prediction can be regarded as a standard time series prediction problem. Traditional methods like autoregressive integrated moving average(ARIMA) [2], long short-term memory neural network (LSTM) [9, 10, 17], Gated Recurrent Unit [18] and temporal convolutional network [21] are used to predict air quality. However, as shown in Fig. 1, air pollutants are not only trend and seasonal in nature, but also related to other influencing factors such as weather and multiple pollutants. It is not comprehensive enough to predict air quality only by the temporal correlation of time series.

Second, several attempts have been made to predict air quality from multiple dimensions [6, 13, 19, 20] including time and factor. These methods usually take the hidden information output from one dimension module as input to another dimension module, or weight and fuse the outputs of several unique dimension modules. However, it is necessary to take into account the fact that air quality factors have different influence degrees in different time periods. As shown in Fig. 1, the Spearman correlation coefficients of meteorology and traffic pollution in the first time period are 0.1556 and 0.1716, respectively, and they are changed to 0.1872 and 0.1321 in the second time period. Therefore, if multidimensional hidden information is not considered at each prediction time step, the multistep prediction results will be greatly compromised.

To address these challenges, in this paper, we propose a cross-correlation gaining framework for predicting urban air quality such as PM2.5, entitled C^2-Guard. First, a temporal correlation module based on Encoder-Decoder unit is utilized to learn the long temporal dependency of air pollutant time series by encoding the air pollutant values on historical time slots. Second, a novel factor correlation module is designed to extract and recalibrate the correlations among air quality-related factors. Multivariate air quality time series are inputted through different channels, and interdependence among different factors is learned through this module. Finally, a cross gaining module is employed to learn the cross-correlations between time and factor dimensions. Joint representation learning is applied to obtain the cross representation at each prediction time step, aiming at reducing the error accumulation of multistep prediction. The main contributions of this work are summarized as follows:

- We propose a novel air quality prediction framework named C^2-Guard that models the complex correlations of temporal features and factor features of air pollutants to predict multistep air quality.
- To learn the correlations of time and factor, a factor correlation module is developed to extract and recalibrate the correlations of related factors affecting air quality. What's more, factor correlations are jointly learned with temporal correlations to gain cross-correlations in a novel and effective cross gaining module.
- Comprehensive experiments are conducted on two real-world air quality datasets. These results indicate that our C^2-Guard performs better than state-of-the-art methods in terms of RMSE and MAE.

The remaining part of the paper is organized as follows. Section 2 discusses the related works. Section 3 formulates the problem of air quality prediction. Section 4 describes our model C^2-Guard. Section 5 presents the evaluation results. In Section 6, we conclude this paper and talk about future work.

2 Related Work

Air quality prediction has always been a hot topic in society. First, air quality prediction can be regarded as a standred time series prediction problem. Conventionally, some classic time series forecasting methods like ARIMA [2] and variants of the recurrent neural network including LSTM [9] and GRU [18] are developed for air quality prediction and other tasks [22, 23]. As a variant of a convolutional neural network, TCN [3] has a flexible receptive field and a stable gradient with good performance in time series modeling. Jorge *et al.* attempt to use TCN to evaluate air quality levels [21]. Ong *et al.* propose a deep recurrent neural network (DRNN) for air pollution prediction by using the auto-encoder model as a novel pre-training method [15]. However, these methods take only historical time data as input, and it has been widely recognized that air pollutants are related to other influencing factors.

To solve this problem, in recent years, several efforts have been made to introduce other air quality factors to enhance prediction performances. Li *et al.* propose a spatiotemporal deep learning (STDL) based air quality prediction method that inherently considers spatial and temporal correlations [12]. Qi *et al.* develop a general and effective approach to improve the performance of the interpolation and the prediction [16]. Zheng *et al.* consider meteorological data, weather forecasts, and air quality data of the station and that of other stations within a few hundred kilometers [19, 20]. Du *et al.* propose a novel deep learning framework named DAQFF [6] that is the state-of-the-art air quality (mainly PM2.5) forecasting method and outperforms the above methods in real datasets. We observe that these methods mainly study the spatiotemporal modeling of air quality and all of them have local receptive fields of factors, so the learned relationships of factors are not comprehensive enough. In addition, temporal correlation information and factor correlation information are in different dimensions. If the information of two dimensions is not simultaneously considered in each prediction time step, the accuracy of multistep prediction will be affected. To the best of our knowledge, our C^2-Guard is the first to learn the correlations of time and factors at each prediction time step in multi-step prediction, thus reducing prediction errors.

3 Problem Formulation

Before formulating the problem of air quality prediction, some necessary mathematical notations are given first. Suppose an urban air quality monitoring station S can detect I types of relevant factors that affect air quality, defined as $\boldsymbol{S} = \{s_1, ..., s_j, ..., s_I\}$. Given an urban air pollutant s_i and a time window with a length of L, a vector $\boldsymbol{s_i} = [s_i^1, ..., s_i^j, ..., s_i^L]$ is defined as the historical values of air pollutant s_i.

Air quality Prediction Based on the above notations, the prediction process is formally defined as follows. Given an air quality factor matrix $X \in N^{I \times T_1}$, where N represents the number of data samples, I represents channels composed of factors that affect air quality and T_1 represents the length of a historical time window. Our task is to learn a predictive model $M : X \rightarrow Y$ from historical air quality factor matrix $\boldsymbol{X}$ to future air pollutant time series $\boldsymbol{Y} = [y_1, ..., y_i, ..., y_{T_2}]$ where T_2 means the prediction range of future air pollutant values.

4 The Proposed Framework

In this section, the proposed framework for the task of air quality prediction is described. The architecture is presented in Fig. 2 where three modules work together. The inputs of temporal correlation module and factor correlation module are historical air pollutant time series and air quality impact factor matrix $\boldsymbol{X^0}$ (shown in the upper part of Fig. 2), respectively from real data. Through the above two modules, the temporal correlations of air pollutants and air quality factor correlations can be modeled. The generated temporal hidden correlation matrix $\boldsymbol{T_{out}}$ and factor hidden correlation vector $\boldsymbol{F_{out}}$ are then fed into the cross gaining module for integration at each prediction time step (shown in the lower right part of Fig. 2). The final output of our framework is future air pollutant values (shown in the lower right corner of Fig. 2). The details of different modules are described in the following sections.

4.1 Temporal Correlation Module

As shown in the upper left corner of Fig. 2, the original input of the temporal correlation module is historical air pollutant time series that are extracted from

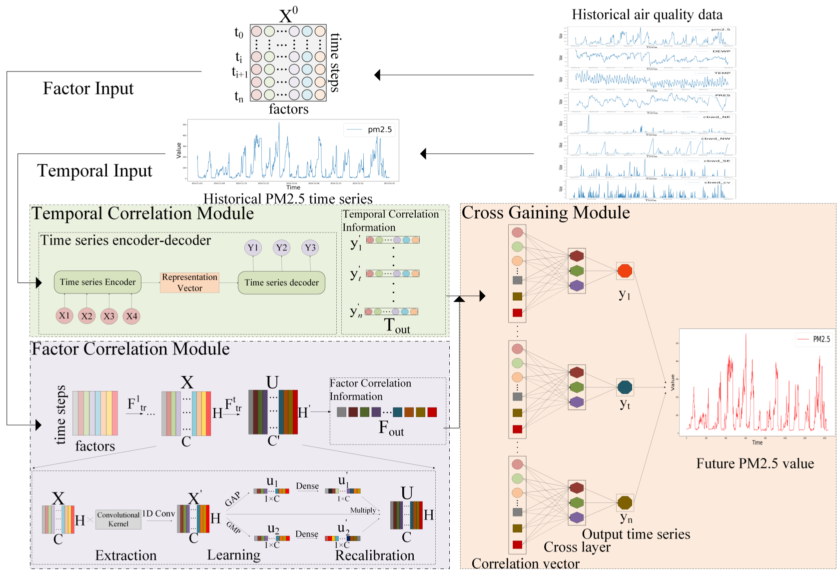


Fig. 2. The overall architecture of the proposed air quality prediction framework C^2-Guard. The prediction of PM2.5 is taken as an example. The upper part is the factor input and the temporal input that come from historal air quality data. The left parts are temporal correlation module and factor correlation module respectively. The right part is cross gaining module that outputs the future PM2.5 data.

the time series of various factors affecting air quality detected at an air monitoring station. To predict multistep time series, we adopt Enc-Dec units (shown in the temporal correlation module of Fig. 2) to learn the long-term temporal correlations of historical air pollutant time series. Since the main work of this paper is to learn the interrelation between air quality factors (discussed in Section 4.2) and how to effectively integrate the temporal correlations and factor correlations (discussed in Section 4.3), the encoding and decoding unit in this module is LSTM that can be replaced by GRU, Transformer and so on.

The final output of this module is a 2-dimension temporal correlation matrix $\boldsymbol{T_{out}}$.

$$\boldsymbol{T_{out}} = [\boldsymbol{y'_1}, ..., \boldsymbol{y'_t}, .., \boldsymbol{y'_n}] \tag{1}$$

where $\boldsymbol{y'_t}$ represents the hidden temporal correlation vector at time step t. n represents the prediction range of air pollutant values. The temporal hiding information at each time step generated in this module will be fused with the factor hiding information in subsequent modules.

4.2 Factor Correlation Module

Essentially, the factors affecting air quality are interdependent, hence the air pollutant to be predicted is related to other air quality factors (shown in Fig. 1). In this part, different factors are input through different channels to form the factor matrix $\boldsymbol{X^0}$ as shown in the upper left corner of Fig. 2. This module explicitly models and recalibrates the interdependencies between factors by superimposing novel convolution and recalibration block F_{tr}^t that is shown in the lower-left part of Fig. 2. In the following content of this section, we describe each part of F_{tr}^t to explain why this block can learn and recalibrate the interdependence between factors.

Factor Correlation Extraction First, a convolutional neural network is used to extract factor correlations of the input factor matrix X. By setting the size and number of one-dimensional convolution kernels, the hidden information between the factors of X will be extracted. The computing process is formalized in Eq. 2.

$$\boldsymbol{X'} = BatchNormalization(Conv1d(\boldsymbol{X})) \tag{2}$$

where $\boldsymbol{X}$ denotes the input factor correlation matrix. $Conv1d()$ refers to one-dimensional convolution. And $BatchNormalization()$ layer is used to reduce overfitting and the insensitivity of the network to the initialization weights. The relationship of different factors will be extracted by this operation. Next, the generated new factor correlation matrix $\boldsymbol{X'} \in \mathbb{R}^{H \times C}$ is inputted into the following block.

Factor Correlation Learning Second, in order to learn the factor correlations effectively, we need to integrate the temporal information in each factor dimension of $\boldsymbol{X'}$ to expand the local receptive field. For aggregating temporal information, Hu *et al*. proposed the use of GAP (GlobalAveragePooling) methods to shrink dimensions [7]. As another pooling method, GMP(GlobalMaxPooling) can also gather temporal information by selecting the maximum value of historical time series. It is reliable in terms of shrinking temporal dimensions. Thus, we use GAP and GMP to shrinking temporal information in each factor channel. Formally, two vectors ($\boldsymbol{u_1}$, $\boldsymbol{u_2}$) are generated by shrinking $\boldsymbol{X'}$ through its temporal dimension H. $\boldsymbol{u_1} \in \mathbb{R}^{1 \times C}$ and $\boldsymbol{u_2} \in \mathbb{R}^{1 \times C}$ are calculated by:

$$\begin{aligned}\boldsymbol{u_1} &= F_{GAP}(\boldsymbol{X'}) \\ &= [\frac{1}{H}\sum_{i=1}^{H}\boldsymbol{X'_{C_1}}(i), ..., \frac{1}{H}\sum_{i=1}^{H}\boldsymbol{X'_{C_j}}(i), ..., \frac{1}{H}\sum_{i=1}^{H}\boldsymbol{X'_{C_n}}(i)]\end{aligned} \tag{3}$$

$$\begin{aligned}\boldsymbol{u_2} &= F_{GMP}(\boldsymbol{X'}) \\ &= [MAX(\boldsymbol{X'_{C_1}}(i)), ..., MAX(\boldsymbol{X'_{C_j}}(i)), ..., MAX(\boldsymbol{X'_{C_n}}(i))]\end{aligned} \tag{4}$$

where C_j refers to the jth factor channel of factor matrix $\boldsymbol{X'}$.

In order to make use of the information summarized in the shrinking opreation, we follow $\boldsymbol{u_1}$ and $\boldsymbol{u_2}$ with an excitation block (two nonlinear fully connected layers) to fully capture factor correlations.

$$\boldsymbol{u'_1} = F_{ex}(\boldsymbol{u_1}, \boldsymbol{W}) = \delta(\boldsymbol{W_2}\delta(\boldsymbol{W_1}\boldsymbol{u_1})) \tag{5}$$

$$\boldsymbol{u_2'} = F_{ex}(\boldsymbol{u_2}, \boldsymbol{W}) = \delta(\boldsymbol{W_2}\delta(\boldsymbol{W_1}\boldsymbol{u_2})) \tag{6}$$

where $\boldsymbol{W_1} \in \mathbb{R}^{\frac{C}{r} \times C}$ and $\boldsymbol{W_2} \in \mathbb{R}^{C \times \frac{C}{r}}$ are weight matrixs of the two layers. To reduce the complexity of the block, the dimensional reduction rate r is set in the first fully connected layer. δ refers to the ReLU [14] function.

The outputs $\boldsymbol{u_1'} \in \mathbb{R}^{1 \times C}$ and $\boldsymbol{u_2'} \in \mathbb{R}^{1 \times C}$ of this block learn and amplify the relationship between different factors.

Factor Correlation Recalibration Third, we multiply the learned relationship vectors $\boldsymbol{u_1'}$ and $\boldsymbol{u_2'}$ with the intermediate factor matrix $\boldsymbol{X'}$ to obtain a new factor matrix $\boldsymbol{U}$. The calculation process is as follows:

$$\boldsymbol{U} = F_{mul}(\boldsymbol{X'}, \boldsymbol{u_1'}, \boldsymbol{u_2'}) = \boldsymbol{X'}\boldsymbol{u_1'}\boldsymbol{u_2'} \tag{7}$$

The new factor matrix $\boldsymbol{U}$ has the same size as the intermediate factor matrix $\boldsymbol{X'}$, and the correlations between factors have been learned and enlarged.

In order to recalibrate the learned factor hiding information with the temporal hiding information at each prediction time step, we need to squeeze the factor matrix U:

$$\begin{aligned} F_{out} &= [\frac{1}{H'}\sum_{i=1}^{H'}\boldsymbol{U}_{\boldsymbol{C_1'}}(i), ..., \frac{1}{H'}\sum_{i=1}^{H'}\boldsymbol{U}_{\boldsymbol{C_j'}}(i), ..., \frac{1}{H'}\sum_{i=1}^{H'}\boldsymbol{U}_{\boldsymbol{C_n'}}(i)] \\ &= [c_1, ..., c_n] \end{aligned} \tag{8}$$

By squeezing the temporal information of each factor channel C_j of $\boldsymbol{U}$, the final factor correlation information vector $\boldsymbol{F_{out}} = [c_1, ..., c_n]$ represents the hidden correlation informations between the factors of the original input matrix $\boldsymbol{X^0}$.

4.3 Cross Gaining Module

In order to gain the cross-correlation at each prediction time step, cross gaining module is proposed to combine the hidden information of the two above modules.

Temporal correlations and factor correlations are fed into the joint layers that combine the correlations of temporal information and factor information into the common space to obtain more accurate prediction results.

$$C_{out} = Joint(T_{out}, F_{out}) = Joint([y_1', ..., y_t', .., y_n'], F_{out}) = \begin{bmatrix} y_1' & F_{out} \\ .. & .. \\ y_t' & F_{out} \\ .. & .. \\ y_n' & F_{out} \end{bmatrix} \tag{9}$$

$$Y = W_{out}'\delta(W_{out}C_{out}) = [y_1, ..., y_t, ..., y_n] \tag{10}$$

The learned factor hidden information vector T_{out} and the temporal hidden informatin vector F_{out} are concatenated at each prediction time step shown as Eq. 9. To gain the final prediction vector, two fully connected layers are used to process the mixed hidden vector at each time step defined as Eq. 10. δ refers to the ReLU function. As shown in the lower right part of Fig. 2, the final output $Y = [y_1, ..., y_t, ..., y_n]$ represents the predicted value of PM2.5 at n time steps in the future.

5 Experiment

5.1 Datasets

Our experiments are based on two real public datasets from UCI[4]. The details of the two experimental datasets are given as follows:

Beijing PM2.5 Dataset [5]This hourly dataset contains the PM2.5 data of US Embassy in Beijing and other related data including meteorological data, wind speed, and so on. The time period of this dataset is between 1/1/2010 to 12/31/2014, and it has 43824 records.

The Temple Of Heaven Air Quality Dataset [6]This dataset includes hourly air pollutants data from the temple of heaven air-quality monitoring site, where the data items include PM2.5, PM10, SO2, NO2, CO, O3, and meteorological data. The time period is from March 1st, 2013 to February 28th, 2017.

5.2 Experimental Setup

Baselines The following three categories of baselines are used to compare with our proposed C^2-Guard: (*i*) the classical machine learning time series prediction methods (i.e., ARIMA); (*ii*) the traditional sequence modeling neural network (i.e., LSTM, GRU, TCN) based on encoder-decoder structure; (*iii*) the state-of-the-art PM2.5 prediction methods(i.e., DAQFF) which learn the interdependence of multivariate air quality-related time series data. More details are listed as follows:

– **Auto-Regressive Integrated Moving Average (ARIMA)**: It is one of the most common statistical models used for time series forecasting and also used in air quality prediction [2, 5]. Scikit-learn is used to build this model in expriment.
– **Long Short-Term Memory Network (LSTM)**: LSTM is widely used in time series prediction tasks and has proposed to predict air quality [9, 17]. A one-layer LSTM network with 100 hidden units is built in our exprements.
– **Gated Recurrent Unit (GRU)**: As an effective variant of LSTM, GRU [18] is implemented in Keras and set the parameters as the same as LSTM.

[4] https://archive.ics.uci.edu/ml/index.php
[5] https://archive.ics.uci.edu/ml/datasets/Beijing+PM2.5+Data
[6] https://archive.ics.uci.edu/ml/datasets/Beijing+Multi-Site+Air-Quality+Data

- **Temporal Convolutional Network (TCN)** [3]: TCN is a popular convolutional neural network capable of processing time series data and widely used to perform fine-grained action segmentation or detection [11] and predict air quality [21]. We utilize the implementation version that is released in GitHub[7] and set the filters and kernel_size are set to 100 and 2.
- **Deep Air Quality Forecasting Framework (DAQFF)** [6]: DAQFF is a state-of-the-art air quality prediction model that consists of Bi-LSTM layers and convolution layers. We implement DAQFF in Keras and set the parameters as the same as mentioned in [6].

Evaluation Methodology To evaluate the performance of C^2-Guard, one year's data are used from each of the two datasets. For Beijing PM2.5 dataset, we select ten-month data for training and validation (01/01/2014-10/31/2014) and two-month data for testing (11/01/2014-12/31/2014). For the temple of heaven air quality dataset, the data from January to October 2016 are used for training and validation, the last two months' data are used for testing. In each experiment, we predict the value of PM2.5 in the next 12 time steps. The Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) are adopted as the evaluation metrics. [6]. The adaptived moment estimation algorithm (Adam) [8] is imployed to optimize the parameters.

Implementation Details Experiments are conducted on an NVIDIA GTX 1060 GPU with 8GB memory. We implement our C^2-Guard through Keras based on Tensorflow [1]. The initial learning rate is 0.01. In addition, the number of neurons in the LSTM and the kerner_size of 1D-CNN are 128 and 64, respectively. Finally, the number of neurons in the dense layer is 16.

5.3 Performance Comparison

To analyze the comprehensive performance of our C^2-Guard, multistep PM2.5 prediction experiments are performed on two datasets, and the experimental results are shown in Table 1.

By comparison, it is not difficult to find that our framework C^2-Guard performs better than all the baselines on both evaluation indicators. It should be noted that the error value in the table is the average of all results on the test dataset, and each value represents the sum of the prediction error value in the next 1$\sim$6 hours or 7$\sim$12 hours. Compared to the state-of-the-art air quality prediction model DAQFF, for the first six hours, our framework C^2-Guard improves prediction quality on two data sets by 12.4% and 10.3%, respectively. And for the next six hours, C^2-Guard improves by 7.3% and 7.4%, respectively.

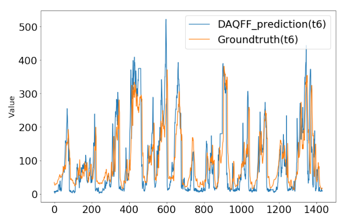
The prediction error at the farther time step is larger because of the forward propagation of the error. To verify that the prediction effect of our framework on various time spectrums in the future is the best, more specific experiments

[7] https://github.com/philipperemy/keras-tcn

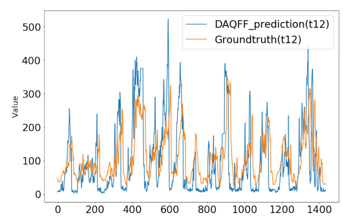
Table 1. In the experiments on two datasets, the errors among different models for the multistep prediction of PM2.5 values in the next 12 hours. The smaller error, the better performance.

Model	Beijing PM2.5 dataset				The temple of heaven air quality dataset			
	RMSE		MAE		RMSE		MAE	
	1h∼6h	7h∼12h	1h∼6h	7h∼12h	1h∼6h	7h∼12h	1h∼6h	7h∼12h
ARIMA [2]	60.3432	84.3432	55.3816	59.3816	58.3432	70.3432	54.3816	66.3816
LSTM [9]	43.0802	63.6328	39.2571	60.5136	44.0545	66.4585	40.0375	63.1094
GRU [18]	40.7278	62.5122	36.7438	59.2638	50.3635	71.1009	46.6650	67.9050
TCN [3]	40.6655	64.7542	34.9552	59.6885	45.9718	71.3350	39.7683	65.8612
DAQFF [6]	40.4987	62.1307	36.5149	58.8346	44.6469	65.9207	40.3688	62.4490
C^2-Guard	**35.4614**	**57.5892**	**31.3690**	**54.1637**	**40.0465**	**61.0232**	**35.6896**	**57.0689**

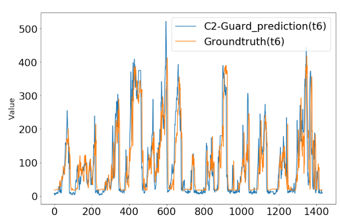
are conducted in the predicted 12 hours. The prediction results at the sixth time point and the twelfth time point in the future are shown in Fig. 3.



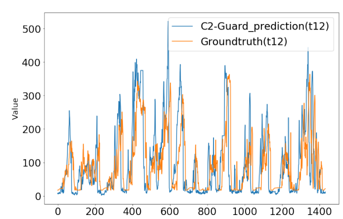
(a) DAQFF(t6)



(b) DAQFF(t12)



(c) C^2-Guard(t6)



(d) C^2-Guard(t12)

Fig. 3. In the experiments on Beijing PM2.5 dataset, a comparison of the sixth hour and the twelfth hour ground truth and predicted PM2.5 value of DAQFF and C^2-Guard. (a) DAQFF for the next 6th hour(t6) prediction; (b) DAQFF for the next 12th hour(t12) prediction; (c) C^2-Guard for the next 6th hour(t6) prediction; (d) C^2-Guard for the next 12th hour(t12) prediction.

Fig. 3 (a)-(d) shows the comparison results of the predicted PM2.5 data and ground truth values of DAQFF and C^2-Guard on the Beijing PM2.5 dataset at different time points (the sixth hour and the twelfth hour). As shown in these figures, the predicted value of our C^2-Guard is closer to the true value at each step. Especially where the extreme value is taken in figures, our predicted value curve is obviously closer to the true value curve, which further reflects the superiority of our model in multi-step prediction. Due to C^2-Guard relies on more time to effectively learn factor correlations, C^2-Guard takes 500s while DAQFF requires 454s in training experiments. It is important to note that the time we spend is within reasonable limits.

5.4 Ablation Study

In this section, an ablation experiment based on two datasets is conducted to gain a better understanding of the effect of factor correlation module. We compare the complete C^2-Guard with the framework without the factor correlation module. The experimental results are shown in Table 2.

Table 2. Comparison of prediction effects of C^2-Guard and C^2-Guard that without factor corrlation module

Model	Beijing PM2.5 dataset				The temple of heaven air quality dataset			
	RMSE		MAE		RMSE		MAE	
	1h∼6h	7h∼12h	1h∼6h	7h∼12h	1h∼6h	7h∼12h	1h∼6h	7h∼12h
C^2-Guard(w/o factor)	38.5199	60.1601	34.5194	56.8523	44.5105	63.0524	39.9345	59.5993
C^2-Guard	**35.4614**	**57.5892**	**31.3690**	**54.1637**	**40.0465**	**61.0232**	**35.6896**	**57.0689**

It can be observed that the errors of the complete C^2-Guard are smaller than C^2-Guard (without factor) on two datasets. Therefore, it is reasonable to learn the correlations of air quality factors.

6 Conclusion and Future Work

In this paper, a novel cross-correlation gaining framework C^2-Guard for urban air quality prediction is explored. This framework, consisting of three modules, performs well on two real datasets. Taking the PM2.5 prediction as an example, experimental results demonstrate that C^2-Guard outperforms the state-of-the-art methods. Some future works are laid in our work. The spatial correlations of air pollution monitoring sites can be added to the learning of our model. Our C^2-Guard is a general framework that can be applied to more prediction tasks.

References

1. Abadi, M., Barham, P., et al.: Tensorflow: A system for large-scale machine learning. In: USENIX Symposium on Operating Systems Design and Implementation. 265–283. (2016)
2. Abhilash, M., Thakur, A., Gupta, D., Sreevidya, B.: Time series analysis of air pollution in bengaluru using arima model. In: Ambient Communications and Computer Systems. 413–426. (2018)

3. Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. CoRR **abs/1803.01271** (2018)
4. Baltrusaitis, T., Ahuja, C., Morency, L.: Multimodal machine learning: A survey and taxonomy. IEEE Trans. Pattern Anal. Mach. Intell. **41**(2), 423–443 (2019)
5. Díaz-Robles, L.A., Ortega, J.C.: A hybrid arima and artificial neural networks model to forecast particulate matter in urban areas: The case of temuco, chile. Atmospheric Environment **42**(35), 8331–8340 (2008)
6. Du, S., Li, T., Yang, Y., Horng, S.J.: Deep air quality forecasting using hybrid deep learning framework. IEEE Trans. Knowl. Data Eng. 1-1(2019)
7. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: CVPR. 7132–7141. (2018)
8. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (Poster) (2015)
9. Kok, I., Simsek, M.U., Özdemir, S.: A deep learning model for air quality prediction in smart cities. In: BigData. 1983–1990. (2017)
10. Krishan, M., Jha, S., Das, J., Singh, A., Goyal, M.K., Sekar, C.: Air quality modelling using long short-term memory (lstm) over nct-delhi, india. Air Quality, Atmosphere & Health **12**(8), 899–908 (2019)
11. Lea, C., Flynn, M.D., Vidal, R., Reiter, A., Hager, G.D.: Temporal convolutional networks for action segmentation and detection. In: CVPR. 1003–1012. (2017)
12. Li, X., Peng, L., Hu, Y., Shao, J., Chi, T.: Deep learning architecture for air quality predictions. Environmental Science and Pollution Research **23**(22), 22408–22417 (2016)
13. Liu, D., Lee, S., Huang, Y., Chiu, C.: Air pollution forecasting based on attention-based LSTM neural network and ensemble learning. Expert Syst. J. Knowl. Eng. **37**(3) (2020)
14. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: ICML. 807–814. (2010)
15. Ong, B.T., Sugiura, K., Zettsu, K.: Dynamically pre-trained deep recurrent neural networks using environmental monitoring data for predicting PM2.5. Neural Comput. Appl. **27**(6), 1553–1566 (2016)
16. Qi, Z., Wang, T., Song, G., Hu, W., Li, X., Zhang, Z.: Deep air learning: Interpolation, prediction, and feature analysis of fine-grained air quality. IEEE Trans. Knowl. Data Eng. **30**(12), 2285–2297 (2018)
17. Tong, W., Li, L., Zhou, X., Hamilton, A., Zhang, K.: Deep learning pm 2.5 concentrations with bidirectional lstm rnn. Air Quality, Atmosphere & Health **12**(4), 411–423 (2019)
18. Wang, B., Yan, Z., Lu, J., Zhang, G., Li, T.: Deep multi-task learning for air quality prediction. In: ICONIP (5). vol. 11305. 93–103. (2018)
19. Yi, X., Zhang, J., Wang, Z., Li, T., Zheng, Y.: Deep distributed fusion network for air quality prediction. In: KDD. 965–973. (2018)
20. Zheng, Y., Yi, X., Li, M., Li, R., Shan, Z., Chang, E., Li, T.: Forecasting fine-grained air quality based on big data. In: KDD. 2267–2276. (2015)
21. Loy-Benitez, J., Heo, S., Yoo, C.: Imputing missing indoor air quality data via variational convolutional autoencoders: Implications for ventilation management of subway metro systems. Building and Environment **182**, 107135 (2020)
22. Vo, N.N., He, X., Liu, S., Xu, G.: Deep learning for decision making and the optimization of socially responsible investments and portfolio. Decision Support Systems **124**, 113097 (2019)
23. Vo, N.N., Liu, S., Li, X., Xu, G.: Leveraging unstructured call log data for customer churn prediction. Knowledge-Based Systems **212**, 106586 (2021)