

## Research on the usability of different machine learning methods in visibility forecasting

Wu WEN<sup>1</sup>, Lei LI<sup>2\*</sup>, P. W. CHAN<sup>3\*\*</sup>, Yuan-Yuan LIU<sup>4</sup> and Min WEI<sup>1</sup>

<sup>1</sup>*School of Computer Science, Chengdu University of Information Technology, Chengdu, 610225, China.*

<sup>2</sup>*School of Atmospheric Sciences, Sun Yat-Sen University, Zhuhai, 519082, China.*

<sup>3</sup>*Hong Kong Observatory, Hong Kong, 999077, China.*

<sup>4</sup>*China Institute of Water Resources and Hydropower Research, Beijing, 100038, China.*

Corresponding authors: \* lilei68@mail.sysu.edu.cn; \*\* pwchan@hko.gov.hk

Received: March 13, 2021; accepted: November 5, 2021

### RESUMEN

La contaminación por neblina, caracterizada principalmente por la baja visibilidad, es uno de los principales problemas ambientales a los que se enfrenta China en la actualidad. El pronóstico exacto de la neblina es útil para la aplicación de medidas preventivas orientadas a controlar emisiones de contaminantes atmosféricos y aliviar la contaminación por neblina. Sin embargo, no es fácil predecir con precisión los acontecimientos de baja visibilidad causados por neblina, lo cual requiere no sólo un pronóstico riguroso de los factores meteorológicos, sino también una actualización detallada y en tiempo real de los inventarios de emisiones de las fuentes. Con el fin de obtener una herramienta de predicción fiable, se estudian varios métodos populares de aprendizaje automático, como la máquina vectorial de apoyo (SVM, por su sigla en inglés), el k-vecino más próximo, el bosque aleatorio y varios métodos de aprendizaje profundo. A partir de los principales factores que influyen en la visibilidad, se analizan las relaciones entre velocidad y dirección del viento, temperatura, humedad y visibilidad. El método de aprendizaje automático se utiliza para entrenamiento y pronóstico. La precisión de estos métodos para pronosticar la visibilidad se verifica mediante varios parámetros (error de raíz cuadrada media, error absoluto medio y error porcentual absoluto medio). Los resultados muestran que: 1) entre todos los parámetros meteorológicos, la velocidad del viento puede reflejar mejor la variación de la visibilidad; 2) el rendimiento de las redes neuronales recurrentes de memoria a largo-corto plazo (LSTM RNN) y el método de unidades recurrentes cerradas (GRU) en la predicción de visibilidad a corto plazo (es decir, 1, 3 y 6 h) es casi igual; 3) el método clásico de aprendizaje automático (SVM) funciona bien para el pronóstico de visibilidad a medio y largo plazo, y 4) el método de aprendizaje automático también tiene cierto grado de precisión en el pronóstico, incluso en un periodo de tiempo más largo (p. ej., 72 h).

### ABSTRACT

Haze pollution, mainly characterized by low visibility, is one of the main environmental problems currently faced by China. Accurate haze forecasts facilitate the implementation of preventive measures to control the emission of air pollutants and thereby mitigate haze pollution. However, it is not easy to accurately predict low visibility events induced by haze, which requires not only accurate prediction for weather elements, but also refined and real-time updated source emission inventory. In order to obtain reliable forecasting tools, this paper studies the usability of several popular machine learning methods, such as support vector machine (SVM), k-nearest neighbor, and random forest, as well as several deep learning methods, on visibility forecasting. Starting from the main factors related to visibility, the relationships between wind speed, wind direction, temperature, humidity, and visibility are discussed. Training and forecasting were performed using the machine learning methods. The accuracy of these methods in visibility forecasting was confirmed through several parameters (i.e., root-mean-square error, mean absolute error, and mean absolute percentage error).

The results show that: (1) among all meteorological parameters, wind speed was the best at reflecting the visibility change patterns; (2) long short-term memory recurrent neural networks (LSTM RNN), and gated recurrent unit (GRU) methods perform almost equally well on short-term visibility forecasts (i.e., 1, 3, and 6 h); (3) a classical machine learning method (i.e., the SVM) performs well in mid- and long-term visibility forecasts; (4) machine learning methods also have a certain degree of forecast accuracy even for long time periods (e.g., 72 h).

**Keywords:** visibility forecast, deep learning, machine learning, time-series forecasting.

## 1. Introduction

Over the last 40 years, with the continuous and rapid economic development of China, the processes of urbanization and industrialization have accelerated, giving rise to environmental pollution problems that have affected especially the air quality. In several Chinese metropolises (e.g., Beijing, Shanghai, and Guangzhou), the urban air quality has continuously deteriorated over an extended period of time. Haze pollution, mainly characterized by low visibility, has become particularly severe and one of the major environmental problems faced by urban areas in China. Since previous research has shown that air pollution is highly correlated to a range of diseases, this problem has drawn considerable public attention. Although great importance has been attributed to the topic and the situation has improved to a great extent, it has not been fundamentally resolved yet. For example, the optimization and adjustment of industrial structures in the Pearl River Delta Region of China in recent years has led to a gradual haze reduction; nevertheless, severe haze episode still occur in case of adverse weather conditions.

Shenzhen is a super city located in the Pearl River Estuary. It is located in the subtropical monsoon climate area and is affected by the alternation of marine and continental monsoon. When the continental monsoon controls Shenzhen, the pollutants emitted from inland dense road networks and factories is transported to Shenzhen by the wind, resulting in air pollution and low visibility.

Accurate haze forecasts facilitate the implementation of preventive measures to control the emission of air pollutants and, thereby, mitigate haze pollution. In addition, the prediction of low visibility events caused by haze is also helpful to traffic management and guidance. Air pollution can be forecasted mainly through numerical and statistical methods (Baklanov et al., 2008; Kim et al., 2010; Gennaro et al., 2013;

Donnelly et al., 2015; Woody et al., 2016; Bray et al., 2017; Zhou et al., 2017). Meteorological and atmospheric chemical models (of numerical forecasting methods) are usually applied to numerically simulate the physical and chemical processes involved in air pollution. Scholars from various countries have accomplished great achievements in this research field, and the large amount of data obtained has been applied in practical operations. Compared with numerical forecasting methods, statistical methods are simpler and require much fewer computing resources. Statistical forecasting models use mainly historical and spatiotemporal samples in haze forecasting to estimate the haze status. Better estimation results can be achieved for short-term haze forecasting. Statistical simulation methods can be divided into two types: linear and non-linear methods.

The linear methods include the autoregressive (AR), autoregressive-moving average (ARMA) and autoregressive integrated moving average (ARIMA) models. In particular, Sharma et al. (2018) applied a time-series regression model to air pollution forecasting, Gupta and Christopher (2009) used a multiple regression approach to monitor air pollution, and Slini et al. (2002) used the ARIMA model to forecast ozone concentrations. Moreover, Deng et al. (2017) established a cellular automation (CA) model based on the multiple regression model to analyze the formation and diffusion of  $PM_{2.5}$ . The computational efficiency of these methods was found to be quite high; nevertheless, the forecasting performance was average in situations involving complex atmospheric changes.

In order to boost the forecast accuracy, many researchers have begun to use non-linear machine learning methods (e.g., support vector machine [SVM], k-nearest neighbor [KNN], and random forest) to conduct studies on air pollution forecasting. In particular, Sotomayor-Olmedo et al. (2013)

applied a support vector machine (SVM) method to air pollution forecasting, Lu and Wang (2005) used an SVM to predict the development trend of air pollution, Dragomir (2010) used the k-nearest neighbor algorithm to forecast the air quality index, and Sun et al. (2016) used the random forest to forecast the air quality index. Non-linear machine learning methods have been widely used for air pollution forecasting, since they can capture well the non-linear characteristics of atmospheric phenomena and have a strong forecasting performance (Pérez et al., 2006; Yildirim et al., 2006; Zhou et al., 2014; Feng et al., 2015; Li et al., 2019).

In recent years, some new deep learning-based forecasting algorithms have been developed rapidly based on previous algorithms, namely deep belief networks (DBN), convolutional neural networks (CNN), and recurrent neural networks (RNN) (Graves and Schmidhuber, 2005; Ren et al., 2017; Tong et al., 2019). In particular, Kuremoto et al. (2014) constructed a DBN for time-series forecasting, Kim (2014) used a CNN to perform sentence-level classification tasks, and Visin et al. (2015) proposed an RNN algorithm for dynamic object recognition. Compared with shallow models, deep learning models can extract the inherent features of deep data correlations. Previous forecasting research has found that deep learning methods have good performances in air pollution forecasting. For example, Huang et al. (2018) proposed a CNN-long short-term memory (LSTM) model for the forecasting of PM<sub>2.5</sub> concentration at different times, and Zhao et al. (2019) predicted PM<sub>2.5</sub> pollution using a long short-term memory-fully connected (LSTM-FC) neural network and monitored the air quality for over 48 h. Additionally, Fan et al. (2017) proposed a spatiotemporal forecasting framework for air pollution based on missing value processing algorithms and a deep recurrent neural network. Each of the above methods, however, has a certain application scope. According to the characteristics of the dataset, we discussed various neural networks of hybrid and integrated models which combine optimization methods with artificial neural networks. Moreover, we performed a short-term forecast of visibility (i.e., one of the main indicators of haze) based on a hybrid neural network model.

The main contributions of this paper are:

1. A discussion and verification of the effects of several factors (e.g. temperature, wind direction and wind speed) in haze forecasting;
2. A comparison of the forecasting performance of several machine learning and deep learning methods commonly used in haze forecasting.
3. An exploration of the validity of these methods through their application to different datasets.

The remaining parts of this paper are organized as follows: section 2 introduces the methodology applied, section 3 establishes the datasets and the experimental design, section 4 describes our analysis of the results, and section 5 presents the conclusions.

## 2. Methodology

### 2.1 Statistical method—the ARIMA model

The autoregressive integrated moving average (ARIMA) model is an analysis method used in time-series forecasting (Aasim et al., 2018; Lai and David, 2020). In the expression ARIMA (p,d,q), AR is the autoregressive term, p the number of autoregressive terms, MA the moving average term, q the number of moving average terms, and d the number of differencing transformations required for series stationarity (order).

Notably, the ARIMA (p,d,q) model is an extension of the ARMA (p,q) model and can be expressed as:

$$\left[1 - \sum_{i=1}^p \varphi_i L^i\right] (1 - L)^d X_t = \left[1 + \sum_{i=1}^q \varphi_i L^i\right] \varepsilon_t \quad (1)$$

where  $L$  is the lag operator,  $d \in Z$ , and  $d > 0$ .

### 2.2 Machine learning methods

In recent years, machine learning has been proven to have a wide range of uses in engineering (e.g., in finance, manufacturing, and retail); nevertheless, it is steadily developing and being introduced into new fields (Khaled and Abdullah, 2016; Doreswamy et al., 2020). At present, the most widely applied machine learning methods include algorithms like the SVM, KNN, and the random forest. The process of machine learning can be summarized as follows: the collected training data are input, all possible hypotheses (expressed as functions) are tested using the learned algorithm, and the hypothesis which is closer to the actual pattern is identified (Liu et al.,

2019, 2020, 2021). This same process is shown in Figure 1.

### 2.3 Deep learning methods

#### 2.3.1 Recurrent Neural Network (RNN)

RNN is a type of recursive neural network which takes sequence data as input and carries out recursions in the direction of the series, with all nodes (i.e., recurrent units) connected in a chain (Feng et al., 2019; Tong et al., 2019). Its network architecture is shown in Figure 2.

The input data were considered to be temporally correlated; moreover, we assumed that  $X_t \in R^{n*d}$  was the small-batch input at time step  $t$  in the sequence  $H_t \in R^{n*h}$  and was the hidden variable at that time step. According to the structure diagram above, the equation of the current hidden variable is as follows:

$$H_t = \varphi(X_t W_{xh} + H_{t-1} W_{hh} + b_h) \quad (2)$$

where  $\varphi$  represents the fully connected layer with the activation function. Through the previous process, the hidden variable  $H_{t-1}$  at the previous time step was saved and a new weight parameter was introduced to describe how the hidden variable of the previous time step was used at the current time step. Specifically, the calculation of the hidden variable at time step  $t$  was done based on both the input at the current time step and the hidden variable at the previous time step. The  $H_{t-1} W_{hh}$  term was added here. From

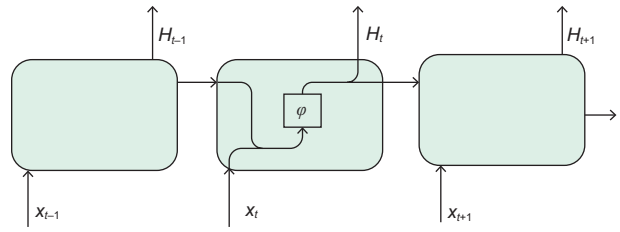


Fig. 2. Network architecture diagram of the recurrent neural network (RNN).

the relationship between the hidden variables  $H_t$  and  $H_{t-1}$  at adjacent time steps in the above equation, it was found that the hidden variable can capture the historical information of the sequence at the current time step. This historical information reflects the state or memory of the neural network at the current time step; therefore, the mentioned hidden variable is also called hidden state. Since the definition of the hidden state at the current time step includes the hidden state at the previous time step, the calculation of the above equation was recurrent. A network with recurrent calculations is called an RNN. Such networks have usually the following activation function:

$$\tanh = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3)$$

#### 2.3.2 Long short-term memory (LSTM)

LSTM is a type of temporal RNN, which was specially designed to solve the long-term dependency problem

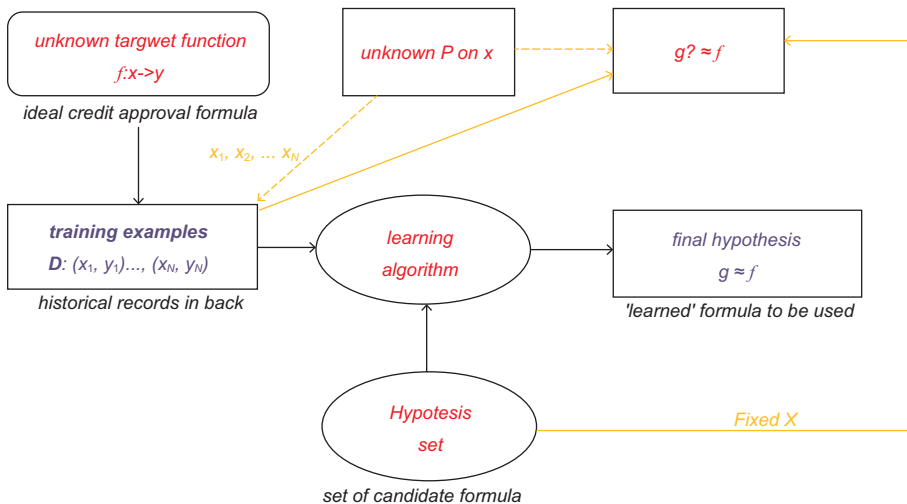


Fig. 1. Schematic diagram of the machine learning process.

of general RNNs (Graves and Schmidhuber, 2005; Yue et al., 2020). All RNNs are structured as chains consisting of repeating neural network modules. The network architecture of the LSTM is shown in Figure 3.

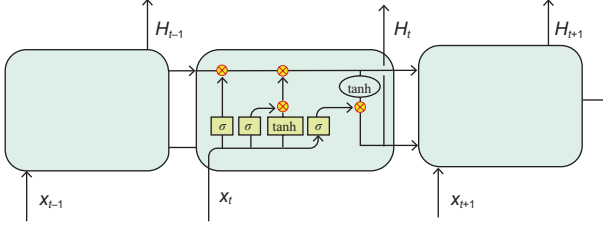


Fig. 3. Network architecture diagram of the long short-term memory (LSTM) model.

An LSTM has the ability to remove or add information to the cell state through carefully designed structures called gates. Each of these structures can selectively pass information; moreover, it contains a sigmoid neural network layer and a pointwise multiplication operator. The LSTM features three gates that are employed to protect and control the cell state. In particular, the forget gate determines what information is discarded from the cell state. Since the output of the sigmoid function is a value  $< 1$ , it is equivalent to a value reduction in each dimension. The input gate determines which new information is added to the cell state; meanwhile, the output gate determines which part of the cell state should be exported through the sigmoid.

### 2.3.3 Gated recurrent unit (GRU)

In order to overcome the inadequacy of RNN in handling long-range dependencies, we proposed the use of an LSTM. A GRU is a variant of the LSTM described above; notably, it is able to preserve the performance of the LSTM, although it is based on a simple structure (Tao et al., 2019). The network architecture of the GRU is illustrated in Figure 4.

From the network architecture diagram, it can be seen that the GRU obtains the states of the two gates from the last transmitted state  $h^{t-1}$  and the input of the current time step  $x_t$ . As shown in Figure 4,  $r_t$  controls the reset gate, which is expressed by the following equation:

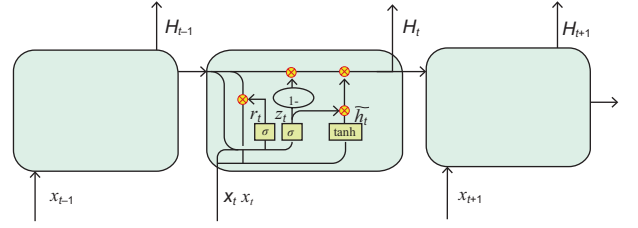


Fig. 4. Network architecture diagram of the gated recurrent unit (GRU) method.

$$r_t = \sigma(W_r^*[h_{t-1}, x_t]) \quad (4)$$

$z_t$  is the update gate, which can be expressed as follows:

$$z_t = \sigma(W_z^*[h_{t-1}, x_t]) \quad (5)$$

where  $\sigma$  is the sigmoid function. Using the function in Eq. (5), data can be transformed into values between 0 and 1 and hence used as gating signals. After obtaining the gating signals, the reset gate can be used to obtain the data after the reset. The reset process is expressed by Eq. (6):

$$\tilde{h}_t = \tanh(W_{\tilde{h}}^*[r_t^*h_{t-1}, x_t]) \quad (6)$$

The  $\tanh$  activation function is mainly used to scale the data to values between  $-1$  and  $1$ . Here,  $\tilde{h}_t$  mainly includes the current input data of  $x^t$  and is added to the current hidden state in a targeted manner. This is equivalent to the memorization of the current state.

Finally, the GRU network performs the forgetting and memorizing steps simultaneously by using the previously obtained update gate  $z_t$ . The updating process is expressed by Eq. (7):

$$h_t = (1 - z_t)^*h_{t-1} + z_t^*\tilde{h}_t \quad (7)$$

The gating signals (i.e.,  $z_t$ ) range between 0 and 1. The closer the gating signal gets to 1, the more data are memorized; meanwhile, the closer it gets to 0, the more data are forgotten. Notably, GRU can simultaneously forget and select memory using only gate  $z_t$ . The output of the final output layer is expressed by Eq. (8):

$$y_t^o = \sigma^*(y_t^i) \quad (8)$$

### 3. Data processing, experimental design, and evaluation

#### 3.1 Data processing

##### 3.1.1 Data overview

In this study, we mainly used relative humidity and visibility data obtained from the Shiyan Observation Base in Shenzhen metropolis between January 1, 2018 and August 31, 2019, as well as the data from the meteorological tower. Though haze is actually closely related to air quality and the Shiyan base does have some observational data on air quality, air quality data is not used in the current study. This is because most weather stations with visibility observations in Shenzhen do not have air quality observational instruments. On the other hand, the purpose of this study is to find appropriate machine learning methods to predict visibility for practical purposes. Therefore, adding air quality data to the current study is not helpful for obtaining a valid conclusion for all weather stations, especially those without air quality observations.

The observation base is located on the east bank of the Pearl River Estuary, about 10 km from the coastline. The visibility data collected from this base between May 2018 and July 2018 were missing and hence replaced by the visibility data collected in the Tiegang Reservoir (around Shiyan base). The sites of Shiyan and Tiegang are actually close to each other (around 5 km apart). By using a dataset with duration of one week from the two stations, it can be found that the correlation coefficient of the data between the two stations exceeds 0.98, and the relative error is less than 5%. The data structure is shown in Table I. All the observation equipment of Shiyan base is calibrated quarterly according to the operational management regulations. Moreover, quality control of the data has been performed and outliers have been deleted before they are used in the current study.

##### 3.1.2 Data pre-processing:

Wind speed and direction, temperature, and humidity data collected continuously from the meteorological

tower were processed as follows, creating a secondary processed dataset for the preparation of the training in the following step:

1. Average wind speed: the average value of wind speed between heights of 10-350 m.
2. Average wind direction: the average value of wind direction between heights of 10-350 m.
3. Average temperature: the average value of temperature between heights of 10-350 m.
4. Average humidity: the average value of humidity between heights of 10-350 m.

The data cycle was 10 min of the monitoring data. The data were normalized where needed and then divided into two subsets: the training and testing data set. Of all data, 80% were used as training data, which included all 2018 data, and the remaining 20% as testing data, which included the data for the first quarter of 2019.

#### 3.2 Experimental design

The flowchart of the experiment is shown in Figure 5. The test data is divided into two parts after data

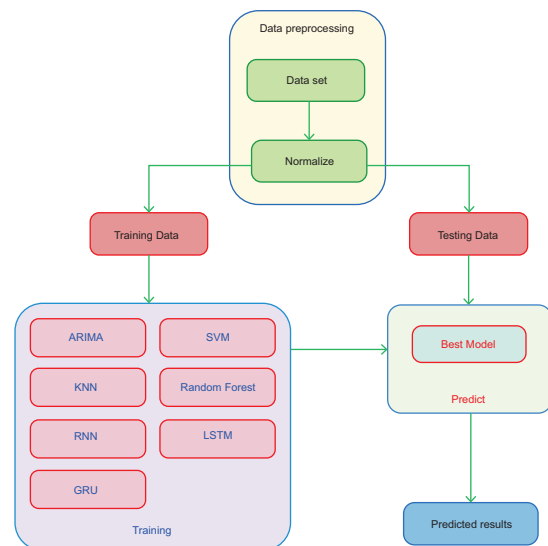


Fig. 5. Experimental flow chart.

Table I. Data information.

Data period	Data type	Height (m)
Training data: 2018/1/1-2018/12/31	Time, wind speed, wind direction, humidity and temperature	10, 20, 40, 50, 80, 100, 150, 160,
Test data: 2019/1/1-2019/3/31		200, 250, 300, 320, 350

normalization. One part is used for training and the other part is used for prediction and verification. Once an optimal method is obtained through the training data, the prediction data will be used to verify the capability of the method.

### 3.3 Evaluation of the results

#### 3.3.1 Root-mean-square error (RMSE)

The RMSE is the square root of the sum of squares of the deviation between the observed and true values and the reciprocal of the number of observations ( $m$ ). This parameter can be used to measure the deviation between the observed and true values. If  $\hat{y}^{(test)}$  represents the predicted value of the model in the testing set, then the RMSE can be expressed as

$$RMSE_{test} = \sqrt{\frac{1}{m} \sum_i \left( \hat{y}^{(test)} - y^{(test)} \right)_i^2} \quad (9)$$

Intuitively, the error drops to 0 when  $\hat{y}^{(test)} = y^{(test)}$ . Moreover,

$$RMSE_{test} = \sqrt{\frac{1}{m} \left\| \sum_i \left( \hat{y}^{(test)} - y^{(test)} \right)_i \right\|^2} \quad (10)$$

Therefore, when the Euclidean distance between the predicted and target values increases, the error also increases.

#### 3.3.2 Mean absolute error

The mean absolute error is the average value of the absolute error, which can better reflect the actual error of the predicted value. If  $\hat{y}^{(test)}$  represents the predicted value of the model in the testing set, then the mean absolute error can be expressed as

$$MAE_{test} = \frac{1}{m} \sum_{i=1}^m \left| \hat{y}^{(test)} - y^{(test)} \right| \quad (11)$$

Intuitively, the error drops to 0 when  $\hat{y}^{(test)} = y^{(test)}$ .

#### 3.3.3 Mean absolute percentage error

The mean absolute percentage error can not only consider the error between the predicted and true values, but also the ratio of the error to the true value. Thus, the mean absolute percentage error was used for evaluating the results. The mean absolute percentage error can be expressed as:

$$MAPE = \sum_{t=1}^n \left| \frac{\text{observed}_i - \text{predicted}_i}{\text{observed}_i} \right| * \frac{100}{n} \quad (12)$$

## 4. Analysis of the results

### 4.1 Factor analysis

Figure 6 shows a small part of the 10-min visibility forecast obtained by using humidity, temperature, wind direction, wind speed, and visibility data as input data. The forecasting results show that the wind speed was the physical parameter which best reflected the trends in visibility (besides visibility itself), followed by relative humidity; however, the forecast error was relatively large for the values of future visibility. It was difficult to predict the results of future visibility merely from temperature and wind direction; on the contrary, it was possible to accurately predict visibility at the next moment from that in the previous moment. Table II shows that the error can be further reduced by adding all factors.

### 4.2 Visibility forecasting performance

In order to compare the advantages and disadvantages of different statistical forecasting methods, we applied the seven methods mentioned in Figure 5 to forecast the visibility in the following 1, 3, 6, 24 and 72 h. The forecasts with different lead time have different functions. For example, the visibility forecast with lead time of 1 to 6 h is of great significance for real-time public traffic dispatching and guidance, while the forecast with lead time of 24-72 h could play a supporting role in longer-term traffic management and pollution source control. By comparing and analyzing the RMSE, MAE, and MAPE statistics of the forecast and actual results, we could determine the most suitable methods for visibility forecasting. In view of the results described in section 4.1, all the physical quantities relative to the previous time step were used for the forecasting of the next time-step data in all seven methods.

The forecasting performances of the seven models are compared in Figure 7, whose left column shows the performances of all seven models while the right column only shows the best one. Table III compares the MAPE of different methods for all forecast experiments, from which the performance of different models can be understood.

As shown in Figure 7a, b, the 1-h average visibility was forecasted using 12 000 training and 3000 testing data points, respectively. The data volume was relatively large. A good forecasting performance could be obtained by all methods except ARIMA, by which

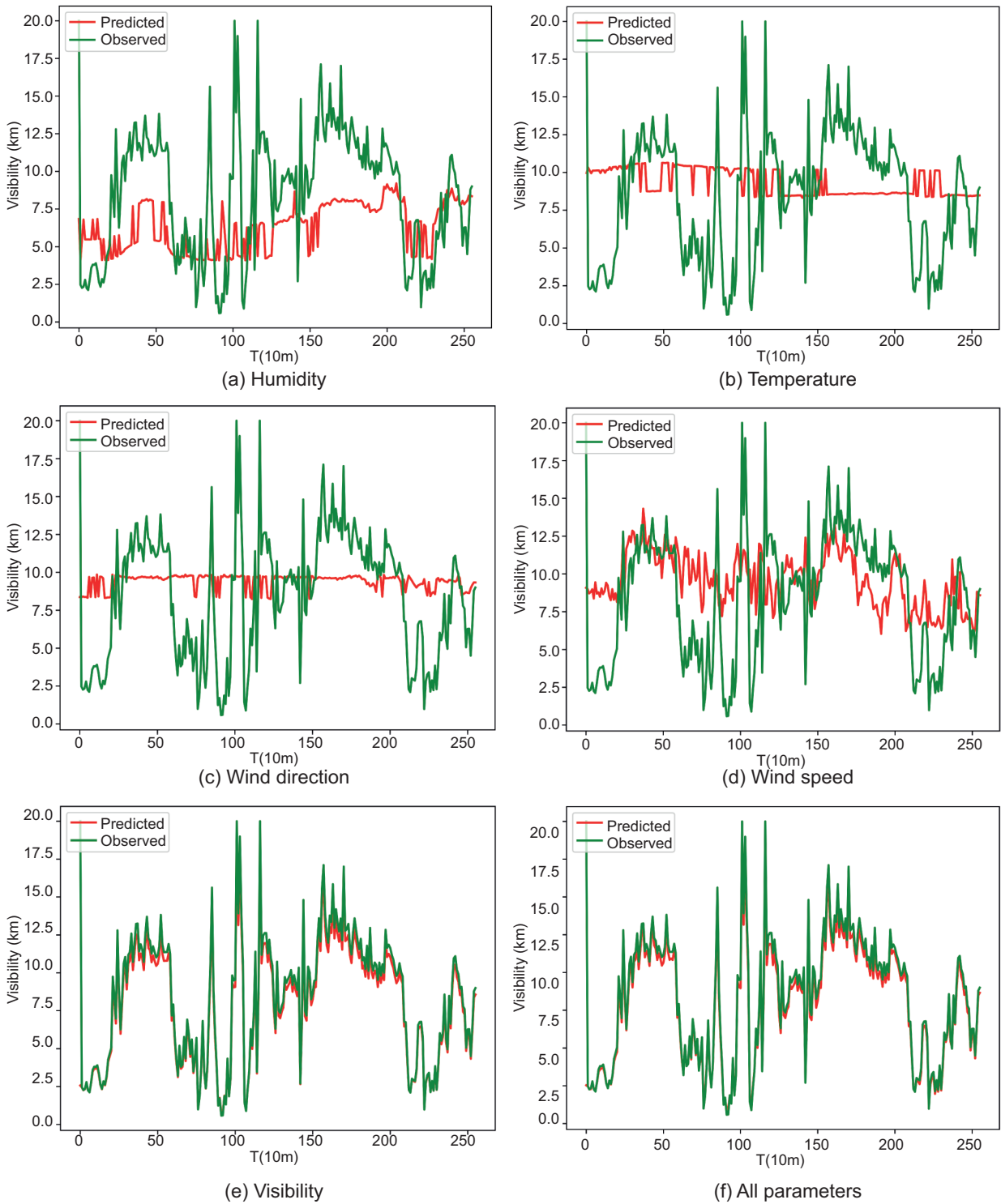


Fig. 6. Forecasting performances obtained by using different meteorological parameters and visibility.

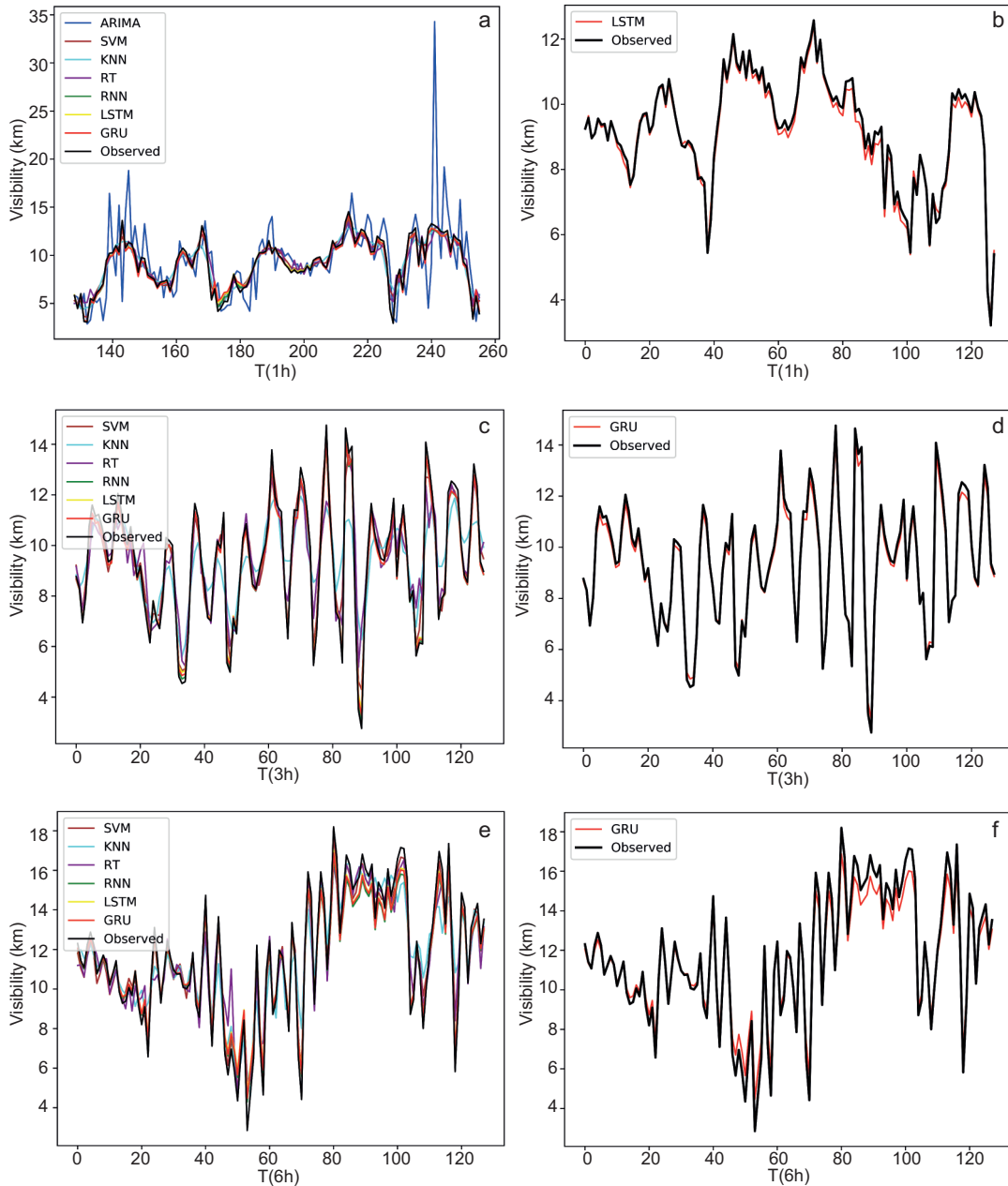


Table II. Multi-factor forecast error table.

Factor	RMSE (m)	MAE (m)	MAPE (%)
Humidity	3905.84	3004.32	27.3
Temperature	3910.16	3060.82	27.82
Wind direction	4197.52	3361.11	30.54
Wind speed	3808.70	2918.81	26.52
Visibility	620.34	550.24	5.01
All parameters	502.33	422.44	3.8

MAPE reached as high as 18.55% in the current study. Given the large errors associated with the ARIMA method compared with those of other methods, it will not be further discussed. In the dataset of 1-h average visibility, the best performance was achieved by the LSTM method, followed closely by RNN and GRU.

In order to learn the performance of the models used in longer lead time, forecast experiments of the 3- and 6-h average visibility were carried out. The



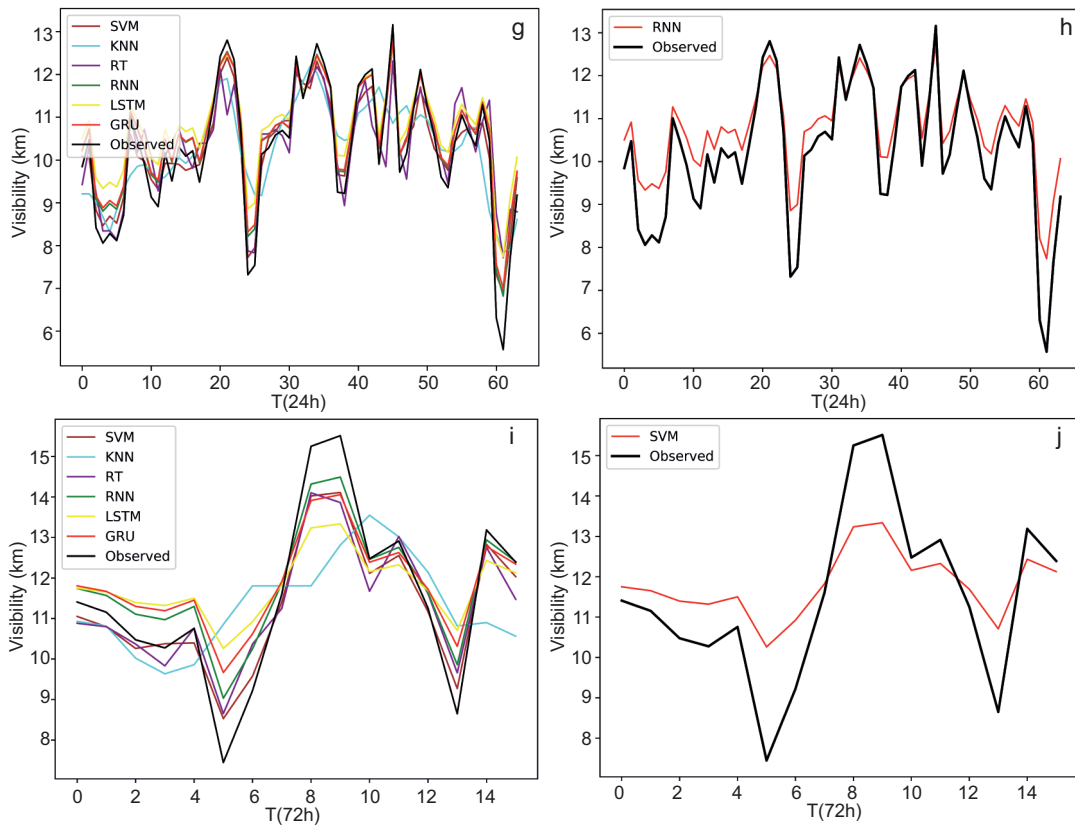


Fig. 7. Comparison of forecasting performances by using different models. The left column is for all seven models and the right column for the best one

Table III. MAPE of different methods for all forecast experiments (%).

Lead time	1 h	3 h	6 h	24 h	72 h
ARIMA	18.55	--	--	--	--
SVM	5.22	4.73	5.39	4.70	5.49*
KNN	7.27	9.46	10.97	9.75	14.8
RT	4.96	6.11	7.92	6.25	6.83
RNN	3.38	3.08	5.66	4.5*	6.75
LSTM	3.12*	2.99	5.62	6.87	11.6
GRU	3.5	2.91*	5.43*	4.83	9.12

\*Best performance.

3-h forecast experiment dataset has 3840 training data points and 960 testing data points, while the 6-h dataset has 1760 training and 340 testing data points. In the two experiments, the best performances were achieved by the GRU method, followed

closely by the RNN and LSTM methods, as shown in Figure 7c-f.

As can be seen in Figure 7g, h, a forecast of the 24-h average visibility was performed using 480 training and 120 testing data points. In this type of dataset, the best performance was achieved by the RNN method, although a similar performance was achieved by the GRU method. A slightly poor performance was observed instead with the LSTM method.

A visibility forecast with a lead time as long as 72-h was also performed using 112 training and 32 testing data points, respectively. Since the number of training datasets was relatively small, the best forecasting performance for this type of dataset was achieved by the SVM method, which provides promising results in case of small-sample time-series forecasting. Despite the deviation in the forecast of the absolute value of visibility, the SVM method could still forecast the trend changes.

## 5. Conclusions

In this study, the visibility forecasting performance of different machine learning methods was compared using visibility data collected in the Pearl River Delta (Shiyan) and observational data obtained from a meteorological tower. The performances of different methods in forecasting the average visibility after 1, 3, 6, 24, and 72 h were tested. The following conclusions were reached:

1. Satisfactory visibility forecasting results could be obtained using historical visibility data. By adding information such as wind speed, wind direction, temperature, and humidity, the forecast accuracy can be further enhanced. In the current study, the MAPE of the prediction could be reduced from 5.01 to 3.8%.
2. Among all meteorological parameters, wind speed was the best at reflecting the visibility change patterns. Though the MAPE obtained by using wind speed is not significantly lower than those obtained by using other parameters, the predicted variation trend obtained by using wind speed is closer to the observed visibility than that obtained by other parameters. Therefore, it can be considered the most important meteorological parameter for future visibility forecasts (apart from visibility itself).
3. When short-term visibility forecasts (i.e., 1, 3, and 6 h) were performed, the forecasting performances achieved by the time-series modeling methods (i.e., RNN, LSTM, and GRU) were comparable to each other, due to the use of a large training dataset.
4. When mid- and long-term visibility forecasts (i.e., of 24 and 72 h) were performed, a classical machine learning method (the SVM) could also provide very good forecasting results, due to the use of a small training data set.
5. The forecasting of visibility using machine learning methods could achieve a certain degree of accuracy even for long periods (e.g., of 72 h). As long as the method selected is suitable, machine learning methods can be practically applied to visibility forecasting, though there is not a universal best method. If only one model is allowed to be used, the RNN is suggested, since the average value of MAPE for all experiments is the smallest one.

Although the machine learning method can achieve satisfactory visibility prediction, the prediction of sudden changes in data trends still has great limitations. Future possible researches to improve the performance could be focused on the following aspects:

1. Decomposing the data, obtaining high- and low-frequency information of the data, and processing different frequency information in different ways to extract the mutation law information in the data.
2. Combining the numerical models and the machine learning methods to improve the accuracy of prediction.
3. Developing new time-series forecasting models to improve forecasting accuracy.

## Acknowledgments

This study was supported by Natural Science Foundation of China (grant number 42075059), and the Science and Technology Projects of Guangdong Province (grant number 2019B121201002).

## References

- Aasim, Singh SN, Abheejeet M. 2018. Repeated wavelet transform based ARIMA model for very short-term wind speed forecasting. *Renewable Energy* 136: 758-768. <https://doi.org/10.1016/j.renene.2019.01.031>
- Baklanov A, Mestayer PG, Clappier A, Zilitinkevich S, Joffre S, Mahura A, Nielsen NW. 2008. Towards improving the simulation of meteorological fields in urban areas through updated/advanced surface fluxes description. *Atmospheric Chemistry and Physics* 8: 523-543. <https://doi.org/10.5194/acp-8-523-2008>
- Bray CD, Battye W, Aneja VP, Tong D, Lee P, Tang Y, Nowak JB. 2017. Evaluating ammonia (NH<sub>3</sub>) predictions in the NOAA National Air Quality Forecast Capability (NAQFC) using in-situ aircraft and satellite measurements from the CalNex2010 campaign. *Atmospheric Environment* 163: 65-76. <https://doi.org/10.1016/j.atmosenv.2017.05.032>
- Deng F, Ma L, Gao X, Chen J. 2017. The MR-CA models for analysis of pollution sources and prediction of PM<sub>2.5</sub>. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 49, 814-820. <https://doi.org/10.1109/TSMC.2017.2721100>

- Donnelly A, Misstear B, Broderick B. 2015. Real time air quality forecasting using integrated parametric and non-parametric regression techniques. *Atmospheric Environment* 103: 53-65. <https://doi.org/10.1016/j.atmosenv.2014.12.011>
- Doreswamy H, Harishkumar KS, Yogesh KM, Gad IM. 2020. Forecasting air pollution particulate matter (PM<sub>2.5</sub>) using machine learning regression models. *Procedia Computer Science* 171: 2057-2066. <https://doi.org/10.1016/j.procs.2020.04.221>
- Dragomir EG. 2010. Air quality index prediction using k-nearest neighbor technique. *Bulletin of PG University of Ploiesti. Series Mathematics, Informatics, Physics.* 42: 103-108.
- Fan JX, Li Q, Hou JX, Feng X, Karimian H, Lin SF. 2017. A spatiotemporal prediction framework for air pollution based on deep RNN. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 4: 15-22. <https://doi.org/10.5194/isprs-annals-IV-4-W2-15-2017>
- Feng R, Zheng HJ, Gao H, Zhang AR, Huang C, Zhang JX, Luo K, Fan JR. 2019. Recurrent neural network and random forest for analysis and accurate forecast of atmospheric pollutants: A case study in Hangzhou, China. *Journal of Cleaner Production* 231:1005-1015. <https://doi.org/10.1016/j.jclepro.2019.05.319>
- Feng X, Li Q, Zhu YJ, Hou JX, Jin Y, Wang JJ. 2015. Artificial neural networks forecasting of PM<sub>2.5</sub> pollution using air mass trajectory based geographic model and wavelet transformation. *Atmospheric Environment* 107: 118-128. <https://doi.org/10.1016/j.atmosenv.2015.02.030>
- Gennaro GD, Trizio L, Gilio AD, Pey J, Pérez N, Cusack M, Alastuey A, Querol X. 2013. Neural network model for the prediction of PM<sub>10</sub> daily concentrations in two sites in the Western Mediterranean. *Science of The Total Environment* 463: 875-883. <https://doi.org/10.1016/j.scitotenv.2013.06.093>
- Graves A, Schmidhuber J. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks* 18: 602-610. <https://doi.org/10.1016/j.neunet.2005.06.042>
- Gupta P, Christopher SA. 2009. Particulate matter air quality assessment using integrated surface, satellite, and meteorological products: Multiple regression approach. *Journal of Geophysical Research: Atmospheres* 114: 1-13. <https://doi.org/10.1029/2008JD011496>
- Huang CJ, Kuo PH. 2018. ReNet: A deep CNN-LSTM model for particulate matter (PM<sub>2.5</sub>) forecasting in smart cities. *Sensors* 18: 2-22. <https://doi.org/10.3390/s18072220>
- Khaled BS, Abdullah K. 2016. Urban air pollution monitoring system with forecasting models. *IEEE Sensors Journal* 16: 2598-2606. <https://doi.org/10.1109/JSEN.2016.2514378>
- Kim YH, Fu JS, Miller TL. 2010. Improving ozone modeling in complex terrain at a fine grid resolution: Part I – examination of analysis nudging and all PBL schemes associated with LSMs in meteorological model. *Atmospheric Environment* 44: 523-532. <https://doi.org/10.1016/j.atmosenv.2009.10.045>
- Kim YH. 2014. Convolutional neural networks for sentence classification. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, Doha, Qatar, 1746-1751.
- Kuremoto T, Kimura S, Kobayashi K, Obayashi M. 2014. Time series forecasting using a deep belief network with restricted Boltzmann machines. *Neurocomputing* 137: 47-56. <https://doi.org/10.1016/j.neucom.2013.03.047>
- Lai YC, David AD. 2020. Use of the autoregressive integrated moving average (ARIMA) model to forecast near-term regional temperature and precipitation. *Weather and Forecasting* 33: 959-976. <https://doi.org/10.1175/WAF-D-19-0158.1>
- Li XL, Luo AR, Li JG, Li Y. 2019. Air pollutant concentration forecast based on support vector regression and quantum-behaved particle swarm optimization. *Environmental Modeling & Assessment* 24: 205-222. <https://doi.org/10.1007/s10666-018-9633-3>
- Liu YY, Li L, Zhang WH, Chan PW, Liu YS. 2019. Rapid identification of rainstorm disaster risks based on an artificial intelligence technology using the 2DPCA method. *Atmospheric Research* 227: 157-164. <https://doi.org/10.1016/j.atmosres.2019.05.006>
- Liu YY, Li L, Liu YS, Chan PW, Zhang WH. 2020. Dynamic spatial-temporal precipitation distribution models for short-duration rainstorms in Shenzhen, China based on machine learning. *Atmospheric Research* 237: 104861. <https://doi.org/10.1016/j.atmosres.2020.104861>
- Liu YY, Li L, Liu YS, Chan PW, Zhang WH, Zhang L. 2021. Estimation of precipitation induced by tropical cyclones based on machine-learning-enhanced analogue identification of numerical prediction. *Meteorological Applications* 28: e1978. <https://doi.org/10.1002/met.1978>
- Lu WZ, Wang WJ. 2005. Potential assessment of the “support vector machine” method in forecasting ambient

- air pollutant trends. *Chemosphere* 59: 693-701. <https://doi.org/10.1016/j.chemosphere.2004.10.032>
- Pérez P, Reyes J. 2006. An integrated neural network model for PM<sub>10</sub> forecasting. *Atmospheric Environment* 40, 2845-2851. <https://doi.org/10.1016/j.atmosenv.2006.01.010>
- Ren SQ, He KM, Girshick R, Sun J. 2017. Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems* 39: 1137-1149. <https://doi.org/10.1109/TPAMI.2016.2577031>
- Sharma N, Taneja S, Sagar V, Bhatt A. 2018. Forecasting air pollution load in Delhi using data analysis tools. *Procedia Computer Science* 132: 1077-1085. <https://doi.org/10.1016/j.procs.2018.05.023>
- Slini Th, Karatzas K, Moussiopoulos N. 2002. Statistical analysis of environmental data as the basis of forecasting: An air quality application. *Science of The total Environment* 288: 227-237. [https://doi.org/10.1016/S0048-9697\(01\)00991-3](https://doi.org/10.1016/S0048-9697(01)00991-3)
- Sotomayor-Olmedo A, Aceves-Fernández MA, Gorrostieta-Hurtado E, Pedraza-Ortega C, Ramos-Arreguín JM, Vargas-Soto JE. 2013. Forecast urban air pollution in Mexico City by using support vector machines: A kernel performance approach. *International Journal of Intelligence Science* 3: 126-135. <https://doi.org/10.4236/ijis.2013.33014>
- Sun HW, Gui DW, Yan BW, Liu Y, Liao WH, Zhu Y, Lu CW, Zhao N. 2016. Assessing the potential of random forest method for estimating solar radiation using air pollution index. *Energy Conversion and Management* 119: 121-129. <https://doi.org/10.1016/j.enconman.2016.04.051>
- Tao Q, Liu F, Li Y, Denis S. 2019. Air pollution forecasting using a deep learning model based on 1D convnets and bidirectional GRU. *IEEE Access* 7: 6690-6698. <https://doi.org/10.1109/ACCESS.2019.2921578>
- Tong WT, Li LX, Zhou XL, Hamilton A, Zhang K. 2019. Deep learning PM<sub>2.5</sub> concentrations with bidirectional LSTM RNN. *Air Quality, Atmosphere & Health* 12: 411-423. <https://doi.org/10.1007/s11869-018-0647-4>
- Visin F, Kastner K, Cho K, Matteucci M, Courville A, Bengio Y. 2015. ReNet: A recurrent neural network-based alternative to convolutional networks. *arXiv:1505.00393*.
- Woody MC, Wong HW, West JJ, Arunachalamas S. 2016. Multiscale predictions of aviation-attributable PM<sub>2.5</sub> for U.S. airports modeled using CMAQ with plume-in-grid and an aircraft-specific 1-D emission model. *Atmospheric Environment* 147: 384-394. <https://doi.org/10.1016/j.atmosenv.2016.10.016>
- Yildirim Y, Bayramoglu M. 2006. Adaptive neuro-fuzzy based modeling for prediction of air pollution daily levels in city of Zonguldak. *Chemosphere* 63: 1575-1582. <https://doi.org/10.1016/j.chemosphere.2005.08.070>
- Yue SC, Chiao HT, Satheesh A, Huang YP, Tsai YT, Lin KM. 2020. An LSTM-based aggregated model for air pollution forecasting. *Atmospheric Pollution Research* 11: 1451-1463. <https://doi.org/10.1016/j.apr.2020.05.015>
- Zhao JC, Deng F, Cai YY, Chen J. 2019. Long short-term memory-Fully connected (LSTM-FC) neural network for PM<sub>2.5</sub> concentration prediction. *Chemosphere* 220: 486-492. <https://doi.org/10.1016/j.chemosphere.2018.12.128>
- Zhou GQ, Xu JM, Xie Y, Chang LY, Gao W, Gu YX, Zhou J. 2017. Numerical air quality forecasting over eastern China: an operational application of WRF-Chem. *Atmospheric Environment* 153: 94-108. <https://doi.org/10.1016/j.atmosenv.2017.01.020>
- Zhou QP, Jiang HY, Wang JZ, Zhou JL. 2014. A hybrid model for PM<sub>2.5</sub> forecasting based on ensemble empirical mode decomposition and a general regression neural network. *Science of The Total Environment* 496: 264-274. <https://doi.org/10.1016/j.scitotenv.2014.07.051>