A Novel Encoder-Decoder Model based on Read-first LSTM for Air Pollutant Prediction

Bo Zhang, Guojian Zou[#], Dongming Qin, Yunjie Lu, Yupeng Jin, Hui Wang*

Abstract—Accurate air pollutant prediction allows effective environment management to reduce the impact of pollution and prevent pollution incidents. Existing studies of air pollutant prediction are mostly interdisciplinary involving environmental science and computer science where the problem is formulated as time series prediction. A prevalent recent approach to time series prediction is the Encoder-Decoder model, which is based on recurrent neural networks (RNN) such as long short-term memory (LSTM), and great potential has been demonstrated. An LSTM network relies on various gate units, but in most existing studies the correlation between gate units is ignored. This correlation is important for establishing the relationship of the random variables in a time series as the stronger is this correlation, the stronger is the relationship between the random variables. In this paper we propose an improved LSTM, named Read-first LSTM or RLSTM for short, which is a more powerful temporal feature extractor than RNN, LSTM and Gated Recurrent Unit (GRU). RLSTM has some useful properties: (1) enables better store and remember capabilities in longer time series and (2) overcomes the problem of dependency between gate units. Since RLSTM is good at long term feature extraction, it is expected to perform well in time series prediction. Therefore, we use RLSTM as the Encoder and LSTM as the Decoder to build an Encoder-Decoder model (EDSModel) for pollutant prediction in this paper. Our experimental results show, for 1 to 24 hours prediction, the proposed prediction model performed well with a root mean square error of 30.218. The effectiveness and superiority of RLSTM and the prediction model have been demonstrated.

Keywords—encoder-decoder model, recurrent neural networks, long short term memory, air pollutant prediction, deep learning, numerical analysis

1. Introduction

Air pollution has become an increasingly serious problem and has caused widespread concerns around the world [1]. The prediction of air pollutant concentration, or simply air pollutant prediction, plays a significant role in air pollution prevention and environment management [2], therefore it has received great attention recently in the research community and has been recognized as a key challenge in environment management research.

Traditionally, air pollutant prediction has been cast as a time series modelling problem using relevant historical data including meteorological factors (e.g. humidity and temperature) and other pollutant factors (e.g. PM_{10} and SO_2). Many studies have shown that there are complex interactions between these factors in the formation of air pollution [3-7]. Therefore, the characteristics of such complex interactions must be extracted and

B. Zhang, Y. Lu, and Y. Jin are with the College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai, 200234, China (e-mail: zhangbo@shnu.edu.cn, 1000459475@smail.shnu.edu.cn, kingpeng21@163.com)

G. Zou is with The Key Laboratory of Road and Traffic Engineering, Ministry of Education, Tongji University, Shanghai, 201804 PR China. (e-mail: 2010768@tongji.edu.cn)

D. Qin was with the College of Electronic and Information Engineering, Tongji University, Shanghai, 201804 and now is with the 3Clear, Beijing, 100029, PR China (e-mail: qindm@3clear.com)

H. Wang is with the School of Computing, University of Ulster, U.K (e-mail: h.wang@ulster.ac.uk)

# This author contributed equally to this work and should be considered co-first author.

* Corresponding authors. Email: h.wang@ulster.ac.uk.

used in air pollutant prediction. According to related researches, the pollutant concentration is a dynamic and continuous process in the temporal dimension [1-4],[7,8],[15-18],[24-28],[33,34],[36],[38-56]. That is, the pollutant concentration at each moment is related to the previous moment, and it also affects the next moment. Therefore, in the task of pollutant concentration prediction, it is particularly important to effectively combine the characteristics of the pollutants and extract the correlation features of the pollutants and meteorological factors in depth.

In many existing studies of air pollutant prediction in a single city, numerical prediction methods are widely used to predict future pollution states based on historical data. Numerical prediction methods can be categorised as follows: deterministic models based on hypothesis theory and prior knowledge [11], empirical black box models based on analysis of input and output time series [12], statistical models [13], traditional machine learning models with small data samples [14]. These traditional methods have some common advantages: low computational complexity, fast calculation speed, and ease of implementation. However, they also have some problems: (1) The pollutant predictions are based on local historical data and empirical rules (as used in traditional complex models including selection of pollutants), so the predictions are summaries of changes of pollutants based on the historical experience, which are insufficient to represent the complex influence of volatile atmospheric environments. (2) The data processing capacity is limited so it is difficult to extract the long-term sequence characteristics of the pollutant concentration and meteorological data and it is impossible to correlate the predicted values of the pollutant concentration at different times. (3) There is correlation between pollutants and meteorological factors [15], which can be exploited for long-term prediction, useful for the prevention of urban pollution incidents. However, it is difficult to extract such correlation fully by existing methods, thus limiting prediction accuracy over long periods of time. The weaknesses identified above lead to poor performance with most traditional air pollutant prediction methods.

For the above problems in the pollutant concentration prediction task, deep learning technology has brought new solutions and performance improvements. Deep learning technology includes a variety of neural network models. These neural network models have different functions. Some networks have the advantage of extracting time series features, and some have the advantages of extracting spatial features, etc. To date, deep learning models have proved to be the state of the art in spatiotemporal prediction tasks [1-4],[7,8],[15-18],[33-36],[38-44],[46],[48-56]. In particular, the concentration of air pollutants is continuous in the temporal dimension, and the concentration of pollutants changes dynamically with time. Therefore, we can use the advantages of neural networks in extracting time series features to improve the accuracy of pollutant concentration prediction task. Many existing studies on air pollutant prediction have shown that deep learning models have better performance than traditional methods, including the

traditional machine learning algorithms, because deep spatial features and deep temporal features can be learned more accurately [1-4],[7,8],[15-18],[24-28],[33,34],[36],[38-56].

So far, we have mainly introduced the relevant methods of pollutant concentration prediction, including traditional methods and deep learning methods. The advantage of these methods is that they use pollutant data or combined with meteorological data to predict the pollutant concentration. In addition, some of the latest methods combine the pollution and weather data of multiple cities in the region to predict the concentration of pollutants in the target city [7],[38],[40,41]. These methods have added different tricks to improve the performance of pollutant concentration prediction. However, these methods each face some weaknesses. A common weakness of these methods is their ineffectiveness in extracting the temporal correlation information in pollutant concentration and meteorological factors from environmental big data. This means that in the long-term sequence prediction task, the prediction performance of the model is insufficient, that is, the pollutant concentration prediction error is large. Our motivation is to find an effective method to predict the air pollutant concentration of a target city based on related historical data (i.e., meteorological data and air pollution data), considering temporal correlations between different factors.

Our task is to solve the common weakness, extract the long-term serial characteristics of pollutant concentration and meteorological data, and ultimately achieve more accurate prediction of pollutant concentration. In this paper, we present a novel deep learning based air pollutant prediction method. It is an Encoder-Decoder model, named EDSModel, where the Encoder uses Read-first Long Short-Term Memory (RLSTM) and the Decoder uses Long Short-Term Memory (LSTM). (1) LSTM is a type of recurrent neural network (RNN) [15] that is able to predict future values using past ones. LSTM has been shown to be well-suited for time series prediction, with better performance than RNNs which suffer the exploding- and vanishing-gradient problems [20-22]. We use LSTM as the Decoder to continuously predict the concentration of air pollutants over a period of time, based on information stored in the Encoder. (2) RLSTM is a new model proposed in this paper, which is based on LSTM. In order to fully extract the long-term sequence features of pollutant concentration and meteorological data, we need to improve the ability of the LSTM model to extract features in the data encoding stage. The gating units of traditional LSTMs are independent of each other, which may lead to problems of low correlation in the feature extraction process and insufficient feature extraction for long-term sequences. RLSTM improves the traditional LSTM gating units to make the control gates interrelated, thereby improving the ability to extract long-term sequence features (i.e. the semantic information of the time series data extracted by the neural network model). To extract the temporal correlation between pollutant concentration and meteorological data, we use RLSTM as an Encoder to extract long-term sequence features from input data.

The main contributions of this paper are as follows:

1) The read-first method is mainly used to filter data feature information, thereby preventing the influence of input redundant information on feature extraction. Therefore, RLSTM, which uses the read-first method as one of the primary core components, is more suitable for long-term sequence feature extraction;
2) The traditional Encoder-Decoder model has been extended, EDSModel, where the Encoder is constructed by multiple layers of RLSTM units, and it can extract long-term sequence features of the input data as well as, at the same time, the complex correlation features between the pollutant concentration and the meteorological data. EDSModel uses the feature vector extracted by the Encoder, which contains contextual semantic information, as input to the subsequent Decoder;
3) The Decoder consists of multiple layers of LSTM units, which predicts the pollutant concentration in the future period n according to the input of the Decoder at time t;
4) Experiments show that our prediction method (RLSTM) achieves better results than state-of-the-art methods.

2. Related work

According to the characteristics of the prediction methods used in related studies, air pollutant concentration prediction can be fundamentally divided into two major research methods: deterministic and statistical approaches [1-6],[8-10],[14],[18],[24-28], [33,34],[36],[38-56].

The deterministic approach can be applied to a limited set of historical data. However, meteorological principles and statistical approaches are needed to simulate the process of real-time emission, diffusion, transformation, and removal of pollutants based on atmospheric physics and chemical reactions. The model structure of a deterministic prediction method is predefined based on certain theoretical assumptions and prior knowledge. There are several commonly used deterministic methods for air pollutant concentration prediction: comprehensive air quality model with extensions (CAMs), the WRFChem model, nested air quality prediction modeling system (NAQPMS), and the community multiscale air quality (CMAQ) model [5],[6],[10],[23].

The statistical approach does not assume a complex theoretical model. Compared with the deterministic approach, it calculates statistics from complex pollutant concentration data and makes predictions on the basis of the statistics, usually showing better predictive performance than the deterministic approach. According to the type of statistics used, there are two branches of the statistical approach, traditional machine learning methods, and new deep learning methods. Traditional machine learning methods include support vector machine [14], multi-label classifier based on Bayesian networks [24], support vector regression (SVR) [25], hidden Markov model (HMM) [26], multiple linear regression (MLR) [18], XGBoost approach [45], and others [47,50]. In recent years, deep learning technology has excelled in dealing with regression

problems, and various neural network based deep learning models have also been applied to improve air pollution concentration prediction performance. Typical network models include multi-layer perceptron (MLP) [27], artificial neural network(ANN) [8],[49], back propagation neural network (BP) [28], RNN neural network [3],[51], LSTM neural network [15],[42-44],[46],[48], deep CNN-LSTM model [7],[38], Gated Recurrent Unit (GRU) [39], Convolutional Long Short-Term Memory (ConvLSTM) [40], and attention-based neural networks [41]. Since air pollutant emissions, diffusion, conversion, and removal are a dynamic process over time, RNN is perhaps a good choice as it can process the time series prediction problem and easily extract temporal features of pollutant concentrations. Therefore, in previous studies, RNN has been used to predict the concentration of air pollutants. Nevertheless, the RNN has two shortcomings: long-term dependencies in input sequences cannot be captured; and a longer interval of input, or the number of RNN layers being larger, may cause a vanishing gradient or exploding gradient problems.

To solve these problems with RNN, Hochreiter and Schmidhuber presented an LSTM neural network model in 1997 [29]. The gate units in LSTM can selectively store information in the input sequence to capture the correlation between the long-term sequence data while solving the vanishing gradient problem. In recent years, LSTM has been successfully applied to many time series prediction problems, such as forecasting the daily maximum price of stocks, machine translation, and speech recognition [30-32]. Moreover, LSTM is also widely used in air pollution prediction tasks, that is, using historical monitoring data of the city to predict the pollutant concentration [15],[42-44],[46],[48]. However, these prediction methods based on the LSTM network mainly predict the pollutant concentration at a certain time in the future, and do not make full use of multiple features in the pollutant data. At present, the pollutant prediction method based on LSTM network faces two key problems. (1) The gate units within LSTM are independent, so LSTM does not efficiently extract the long-term sequence characteristics of the input data [53],[59,60]. (2) Existing LSTM pollutant prediction models can only predict pollutant concentration at one time, and cannot accurately predict the concentration of pollutants in the future, that is, it is difficult to correlate the predicted values at each time [42-44].

In our research, we argue that if we want to make an accurate prediction of air pollution concentration for a future period, we must build on the historical observation data. The above researches are generally based on the shallow feature extraction of pollutants and meteorological data, including two aspects: spatial dimension and temporal dimension. From the spatial dimension, for multi-site pollutant concentration prediction tasks, some researchers have used CNN to extract spatial features of pollutant and meteorological data [7],[38],[40,41]. However, for single-site pollutant concentration prediction, from the temporal dimension, they cannot fully extract the time series distribution features of historical data, especially the complex internal interactions between long-term series

data, including deterministic and statistical approaches. Therefore, an Encoder-Decoder prediction model is designed to extract the valid information of historical observation data and reasonably predicts the future concentration of pollutants. The model uses an Encoder-Decoder architecture in which the Encoder captures historical data information and the Decoder predicts the air pollution concentration. To improve the ability of prediction, we enhance the information acquisition ability of the Encoder by modifying an LSTM network structure, so that there are more reference bases in the prediction stage to drive the model to make an excellent judgment.

This paper fully considers that the prediction model should make a more accurate prediction of the air pollution concentration of the target city in the future period of time, and it should accomplish the following objectives: (1) Effective use the city's historical pollutant concentration and meteorological big data; (2) Deep mining of the long-term correlation features of historical pollutant and meteorological data.

3. Improved Long Short-term Memory Network

Air pollutant prediction is a typical time-series prediction problem, i.e., predicting the value of the following period based on the sequence data of a known period. Studies show that LSTM has good performance in air pollutant prediction tasks [15],[19],[42-44],[46],[48]. This paper introduces the Encoder-Decoder model for pollutant prediction. LSTM is modified, resulting in RLSTM, and used as the Encoder. This model aims to solve the problem of long-term dependence and realize time-series prediction. Compared with LSTM, RLSTM can more effectively process time-series data, compress and extract the information of the historical pollutant concentration and meteorological data. It can extract valid information from this data, drive the model to learn the distribution features of historical data and prevent the dispersion of important information, and finally lead EDSModel to predict the pollutant concentration of the following period accurately. The nomenclature used in this paper is shown in Table 1.

Table 1 Nomenclature

	Symbol	Description
General	h	Hidden state
	C	Cell state
	$\cdot$	Matrix multiplication
	$*$	Matrix dot multiplication
	$+$	Matrix addition
	σ	Sigmoid function
	$Tanh$	Tanh function
	W	Weights
	b	Bias
	λ	regularization parameter
RLSTM	f	Forget gate
	r	Read gate
	w	Write gate
LSTM	f	Forget gate
	i	Input gate
	O	Output gate

3.1 Traditional LSTM

LSTM can learn to bridge minimal time lags by enforcing constant error flow through "Constant Error Carrousels"(each memory cell has at its core a recurrently self-connected linear unit called the " Constant Error Carrousels", whose activation we call the cell state) within special units, called cells. Multiplicative gate units learn to open and close access to the cells [19],[29]. The LSTM neural network has a memory cell and a state, and completes state and cell updates by the LSTM gating mechanism. These gates are input gate, forget gate and output gate. Compared with other RNNs, including GRU, LSTM has achieved better performance by adding a gating mechanism to control the flow of information and the update of states and cells [52]. The LSTM network architecture in conjunction with an appropriate gate-based learning algorithm can overcome the well-known exploding- and vanishing-gradient problems that occur during long-term sequence feature extraction [29].

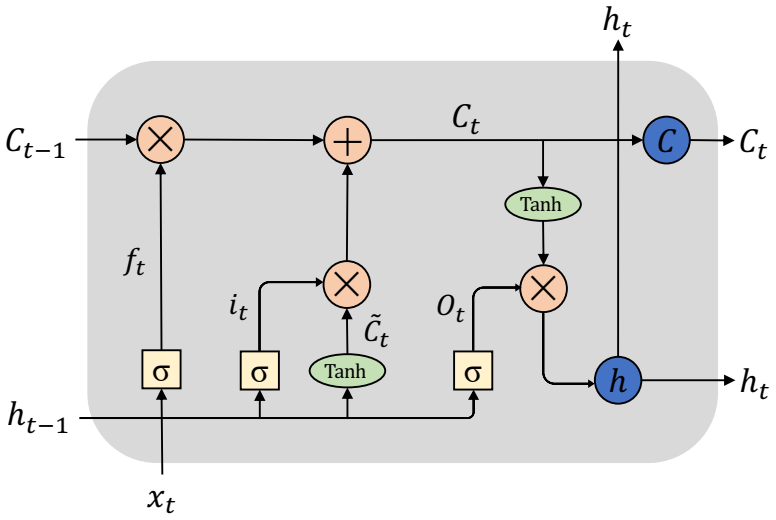


Fig. 1. Traditional LSTM cell. i, f, and O represents input gate, forget gate and output gate, respectively. x is the input feature, h is hidden state and C is cell memory state. '×' and '+' represent the multiplication and addition operations of the matrix, respectively. σ and $Tanh$ are activation functions.

Fig.1 describes the detailed internal gate control units of the traditional LSTM cell unit. It has three gate control units: forget gate, input gate and output gate. The three gate control units are independent of each other, and perform information forget, update, and output operations on the time series features information, respectively [60]. Therefore, during the time series feature extraction phase, the LSTM will encounter the following problems: (1) When forget gate f_t selectively forgetting the cell memory information C_{t-1}, the update information $i_t * \tilde{C}_t$ is not referred to, and the effect of the update information $i_t * \tilde{C}_t$ at time t on the forget of cell memory information C_{t-1} is ignored. (2) At time t, the update of the memory information of cell C_t is mainly completed through the cooperation of the forget gate f_t and the input gate i_t. However, when the input gate i_t selects the information $\tilde{C}_t$ for updating the cell state C_{t-1}, it does not refer to the information forgotten by the forget gate f_t. Therefore, the forget gate i_t and the

input gate f_t are two independent processes in LSTM. (3) Because the forget gate f_t and input gate i_t are independent in the feature extraction process, the hidden feature information h_t output by the output gate O_t may have problems such as feature information redundancy or insufficient feature extraction [19].

In our opinion, the above three problems have seriously affected the ability of LSTM to extract time series features. In the next section we will present an improved LSTM to solve the above problems — Read-first Long Short-Term Memory (RLSTM).

3.2 RLSTM

3.2.1 Feature Extraction Process

In the last section, we have identified the problems with the traditional LSTM. In this section, we will focus on solving these problems by proposing a new model, an improved LSTM. In order to better extract the time series correlation features of pollutant concentration and meteorological data, and to achieve accurate prediction of pollutant concentration, we seek to improve the gate units of traditional LSTM and propose a new LSTM structure, RLSTM, that have new gates – read gate r_t, forget gate f_t and write gate w_t. The working mechanism of RLSTM is: firstly, the read gate selects input data feature information; secondly, the forget gate forgets the historical cell memory information C_{t-1}; finally, the write gate updates the cell memory information C_{t-1}. Its structure is shown in Fig. 2, and the features extraction steps are described below.

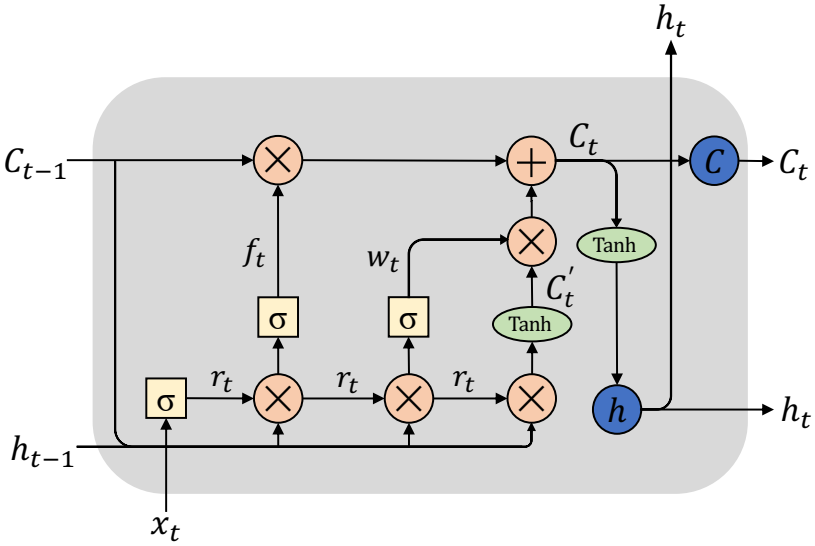


Fig. 2. An RLSTM cell is composed of gate units, memory unit, activation functions and matrix operation. r, f, and w represents read gate, forget gate and write gate, respectively. x is the input feature, h is hidden state and C is cell memory state. '×' and '+' represent the multiplication and addition operations of the matrix, respectively. σ and $Tanh$ are activation functions.

Step 1. We use the read-first method, that is, extracting the features of the input data by the read gate, and appropriately filtering the redundant information in the input data. The input data mainly includes the state of the cell C_{t-1} and the hidden state h_{t-1} of the output at the last moment, and the time series feature value of the current input x_t. The

read gate uses the Sigmoid function as the activation function of the information filter to constrain the range of feature information values. The filter method is similar to the attention mechanism. That is, we use the sigmoid function to weight the input features $[h_{t-1}, x_t, C_{t-1}]$, a small weight indicates that the importance of the feature information is less, and a large weight indicates that the feature is more important [53]. The weighting process is completed through training of the neural network, and the weight ranges are between [0.0,1.0]. For example, $r_t * [h_{t-1}, x_t, C_{t-1}]$ =[0.0,0.4,0.6,1.0]*[1,2,3,4], where feature element 1 may be redundant information, and feature element 4 is important information. This is shown in Equation (1):

$$r_t = \sigma(W_r.[h_{t-1}, x_t, C_{t-1}] + b_r) \quad (1)$$

The read gate uses the hidden state $[h_{t-1}, C_{t-1}]$ at time $t-1$ and the current input x_t, to generate a weight matrix r_t that eliminates redundant information (by matrix multiplication '.' and sigmoid activation function 'σ'), as the basis for updating the unit state at the current time t, and its function is similar to the input gate of traditional LSTM.

Step 2. When updating the state information of the memory unit C_{t-1} at time $t-1$, we need to select important input feature information, which has been completed in step 1; we also need to selectively forget the information in the memory unit C_{t-1}. Therefore, we choose the Sigmoid function as the activation function of the forget gate. The principle of the Sigmoid function here is the same as in the Step 1. Unlike LSTM, RLSTM forgets the feature information with a small contribution in the memory unit C_{t-1} according to the output of the read gate r_t and the current input $[h_{t-1}, x_t, C_{t-1}]$.

$$f_t = \sigma(W_f.(r_t * [h_{t-1}, x_t, C_{t-1}]) + b_f) \quad (2)$$

$$\hat{C}_t = f_t * C_{t-1} \quad (3)$$

The forget gate inputs the calculation result of the logistic regression function $W_f.(r_t * [h_{t-1}, x_t, C_{t-1}]) + b_f$ into the Sigmoid function. The Sigmoid function assigns different weight values to the feature elements in each dimension of the memory unit C_{t-1} according to the logistic regression function value, and the value is constrained between [0.0,1.0]. By multiplying the output of the memory unit C_{t-1} and the forget gate f_t, a part of the memory information is forgotten (if the weight value is 0.0, the corresponding memory information is completely forgotten)

Step 3. After part of the state information of the memory unit is forgotten, RLSTM needs to update the information of the memory unit according to the current read gate r_t and write gate w_t, that is, to perform the memory information write operation. The write gate w_t is based on the output of the read gate r_t and the input $[h_{t-1}, x_t, C_{t-1}]$ at the current time t, and selects some important feature information for updating the information of the memory unit $\hat{C}_t$ at the current time. Therefore, write gate w_t updates the memory unit information based on the important features selected by the read gate r_t, thereby reducing redundant information residuals. This is shown in equations (4), (5) and (6):

$$w_t = \sigma(W_w.(r_t * [h_{t-1}, x_t, C_{t-1}]) + b_w) \quad (4)$$

$$C_t' = Tanh(W_c.(r_t * [h_{t-1}, x_t, C_{t-1}]) + b_c) \quad (5)$$

$$C_t = \hat{C}_t + w_t * C_t' \quad (6)$$

In the above equations, C_t' represents the initial feature used to update the information of the memory unit $\hat{C}_t$. The initial feature is calculated by the output of the read gate r_t and the input $[h_{t-1}, x_t, C_{t-1}]$ of the RLSTM, as shown in equation (5). The implementation principle of the write gate w_t is the same as the forget gate f_t, as shown in equation (4). The function of write gate w_t is mainly to assign different weight values to the elements in each dimension of C_t', and to select the important feature information for updating the memory unit $\hat{C}_t$ (if the weight value is 1.0, the corresponding dimension feature information is all saved). Finally, the memory cell $\hat{C}_t$ after the forget gate f_t operation in step 2 is added to the important feature $w_t * C_t'$ selected by the write gate w_t in step 3, and the result is the unit state C_t output at time t, as shown in equation (6).

Therefore, the characteristics of RLSTM are summarised as follows:

1. It has three gates which are related to each other, and the function is different from LSTM which RLSTM updates memory cell status information with low redundancy; and
2. There are dependencies among its activation functions (σ and $Tanh$).

3.2.2 Advantages of RLSTM

Similar to an LSTM network, RLSTM has the same number of gate units and includes an input layer, hidden layer and output layer. There are forward connections between them: the connection between the input layer and the hidden layer, and the connection between the hidden layer and the output layer. Unlike LSTM, the three gates of RLSTM are read gate r, write gate w and forget gate f, which are different from the gates in LSTM. Specifically, the RLSTM read gate compresses the input data according to the current input and the state of the cell at the previous time, filtering out redundant information in the input data (see section 3.2.1 for detail). Similar to the input gate control of the LSTM, the read gate of the RLSTM controls the amount of information that flows into the cell and filters out useless information. Secondly, the three LSTM gates are independent of each other, whereas the RLSTM write gate and forget gate are calculated on the basis of read gate. The forget gate of RLSTM is similar to the one in LSTM, which selectively forgets the information of the current unit. The write gate controls how much information can be written to the cell of the network at the current time and update the current unit status information. RLSTM uses these three gates to control the inflow and outflow of the cell.

The equations of RLSTM are as follows:

$$\begin{cases} r_t = \sigma(W_r \cdot [h_{t-1}, x_t, C_{t-1}] + b_r) \\ f_t = \sigma(W_f \cdot (r_t * [h_{t-1}, x_t, C_{t-1}]) + b_f) \\ w_t = \sigma(W_w \cdot (r_t * [h_{t-1}, x_t, C_{t-1}] + b_w) \\ C'_t = tanh(W_C \cdot (r_t * [h_{t-1}, x_t, C_{t-1}]) + b_C) \\ C_t = f_t * C_{t-1} + w_t * C'_t \\ h_t = tanh(C_t) \end{cases} \quad (7)$$

where r_t, f_t, and w_t are the three gates mentioned above; x_t is the input of the t-th step, h_{t-1} is the output of the hidden layer at the $(t-1)$-th step, h_t is the output of the hidden layer at the t-th step, and W_r , W_w and W_f are the weight matrix of the read gate, write gate and forget gate, respectively; b_r, b_w and b_f are the bias matrix of the cell; σ is the Sigmoid activation function; $tanh$ is the Tanh activation function.

Compared to traditional LSTM networks, RLSTM is more suitable for extracting temporal correlation features of pollutant concentration and meteorological data. When RLSTM reads the input sequence, the read gate automatically filters out redundant information of the input data. In addition, the forget gate and write gate calculations are performed after reading, using more valuable information to update the state of the cell, thereby causing the cell to abandon redundant and useless 'Memory'. This facilitates the long-distance propagation of the gradient in the reverse direction, preventing the vanishing-gradient and the degradation of the prediction performance as the sequence length increases. In most cases, the concentration of pollutants and the value of meteorological factors change smoothly in a short period of time, and generally do not cause violent fluctuations. Therefore, for long-term pollutant concentration prediction tasks, the traditional LSTM in the limited memory space will cause the model to extract too much redundant feature information of pollutant concentration and meteorological data in a short time [19]. In order to improve the prediction accuracy of the model in the actual pollutant concentration prediction task, the prediction model should extract more temporal correlation features of historical pollutant concentration and meteorological data. Therefore, this paper takes advantage of RLSTM in temporal correlation feature extraction and uses RLSTM as the Encoder of EDSModel for feature extraction.

3.3 State-of-the-art Encoder-Decoder Model

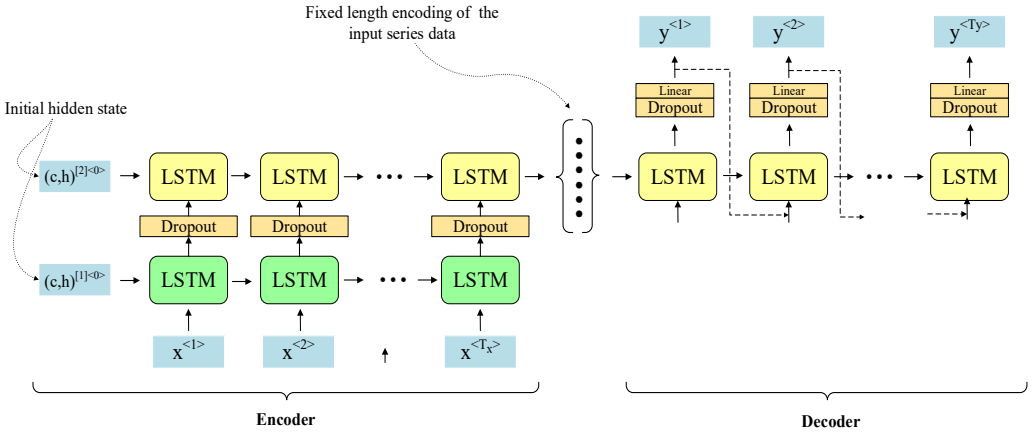


Fig. 3. Encoder-Decoder model. Using stacked LSTMs for encoding and one LSTM layer for decoding.

In computer science and related fields, the Encoder-Decoder model based on recurrent neural network has been widely used in time series prediction tasks [33-36]. Recently, the mainstream pollutant concentration prediction model is mainly based on LSTM

Encoder-Decoder, using one LSTM as the Encoder to extract the timing characteristics of historical data, and then using another LSTM as the Decoder for time series prediction [54-56]. The overall architecture of the model is shown in Fig. 3.

In the Encoder-Decoder Model, the Encoder part compresses the information from the entire input sequence into a vector which is generated from the sequence of the LSTM hidden states. The fixed-dimensional representation of the input sequence is given by the last hidden state of the encoding part as shown in Fig. 3. The decoding part has one LSTM layer for predicting the output sequence. In addition, attention operations can be performed on the hidden vector output by the Encoder at each time to obtain a global weighted feature vector [35,36],[54-56]. However, all of these tasks are based on the characteristics of the LSTM itself or the addition of ancillary feature processing operations based on this, without paying attention to or changing the feature extraction process of the LSTM itself. In the following work, we will continue to use the Encoder-Decoder model for time series prediction of pollutant concentration, but we focus on the LSTM structural transformation and performance of feature extraction.

3.4 EDSModel for Air Pollutants

To prove that RLSTM is more suitable for the extraction of long-term series of pollutants and meteorological data features than LSTM, we choose LSTM as the Decoder of EDSModel in this paper. That is, our purpose is to verify through experiments that when the Decoder is the same, the prediction model based on RLSTM as the Encoder is better than the LSTM as the Encoder. The Encoder and Decoder are connected through the context vector C. The EDSModel proposed in this paper adjusts network hyperparameters and weights through multiple experiments to achieve accurate prediction of pollutant concentrations in the future.

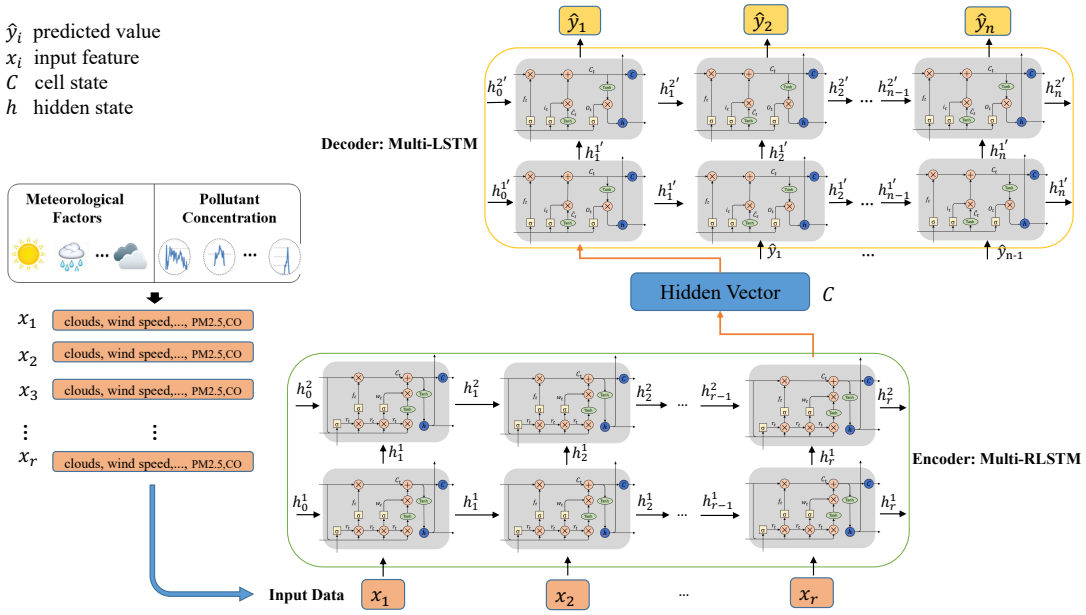


Fig. 4. An EDSModel. Using stacked RLSTMs for encoding and one LSTM layer for decoding.

For example, Fig. 4 shows an EDSModel for a two-layer network structure. The upper part of the figure shows the overall network structure of the EDSModel, the Encoder consists of two layers of stacked RLSTM, and the corresponding Decoder consists of two layers of stacked LSTM. The middle part of the figure shows a layer structure of the EDSModel. The lower part of the figure shows the specific internal structure of RLSTM and LSTM.

The EDSModel process is as follows:

First, the Encoder consists of multiple layers of RLSTM, which sequentially extracts the temporal correlation features of the input pollutant concentration and meteorological data, and finally generates the context feature vector C of the time-series correlation feature information, that is, hidden vector. Then, in the prediction phase of the EDSModel, the Decoder accurately predicts the concentration of pollutants in the future period based on the context feature vector C. At time t_1, the Decoder has no input and should be filled with the all zero values as the signal that the Decoder starts. The Decoder generates the predicted value only from the context vector C at time t_1. At time t_i, the Decoder generates the predicted value through the hidden feature h_{i-1} in combination with the output $\hat{y}_{i-1}$ of time t_{i-1}, and so on. At last, the Decoder produces a time-series pollutant concentration prediction step by step.

The EDSModel proposed in this paper has two parts, RLSTM which is for encoding the input sequence and LSTM which is for decoding the output sequence. To demonstrate the effectiveness of our proposed model, in the experiment section, we choose different types of recurrent neural networks as Encoders and Decoders for predictive performance comparison. Among them, the number of layers of the Encoder and the Decoder can be adjusted.

4. Experimental results

4.1 Data description

The experiment used historical pollutant concentration and meteorological data from monitoring stations in 10 cities collected from May 13, 2014 to May 30, 2018 (Data sample and codes, URL: http://github.com/zouguojian/data). The experimental data in this paper is based on the city level, that is, the sample data of each city every hour is a one-dimensional feature vector, and the feature elements are composed of pollutants and meteorological factors. 10 cities are selected, Shanghai, Nanjing, Hangzhou, Wuhan, Beijing, Shenyang, Harbin, Chengdu, Wulumuqi, and Lasa, which have different economic development in China. The geographical location of these cities is scattered throughout the country, and the pollution environment of the cities varies. We selected 16 pollutants and meteorological factors: AQI, $PM_{2.5}$, PM_{10}, SO_2, NO_2, O_3, CO, Temp (temperature), Hum (humidity), air pressure, wind direction, wind speed, clouds, maximum temperature, minimum temperature and Conds (meteorological conditions). Fig. 5 shows the locations of all monitoring sites.

Fig. 5. Selected cities.

4.2 Experimental Setup

4.2.1 Datasets

In our experiment, we selected 70% of the data as the training set, 15% as validation set, and the remaining 15% was used as the test set. The specific method of dividing the data in this study is as follows: first, we divide the data set uniformly according to a given window length L and a moving step size of S, and finally the total number of samples obtained is $N = ((D - D * 0.15) - L)/S$; then, we scramble the N samples, select 82% of them as the training set and 18% as the validation set. In addition, 15% of D is used as the test set, which means that we extract 15% of the data from the original data set as the test set without disturbing it; finally, we define our division method as a generalized random method. Among them, the window length L represents the sum of the time sequence length of the input model and the target prediction sequence length, and D is the size of the original data set. The missing values of the air pollutant concentration and meteorological data set are filled by spatiotemporal interpolation [37]. This paper attempted to predict the future n hour pollutant concentration in the target city by using the pollutants and meteorological data in the past r hour. $\hat{y} = P(\hat{y}_{t+1}, \hat{y}_{t+2}, \dots, \hat{y}_{t+n} | x_t, x_{t-1}, \dots, x_{t-r+1})$ where $\hat{y}_i$ represents the predicted value and $x_i \in R^m, (m = 16)$represents the observed value.

4.2.2 Related Definitions of EDSModel

In the EDSModel, the loss function is used to measure the degree of inconsistency between the predicted value $\hat{y} = (\hat{y}_1, \dots, \hat{y}_i, \dots, \hat{y}_n)$ and the real value $y = (y_1, \dots, y_i, \dots, y_n)$. The loss function is given in (8):

$$loss = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} + L_2, (L_2 = \frac{\lambda}{2}\|W\|_2^2) \quad (8)$$

where n is the length of the predicted sequence, y_i denotes the observed value of the pollutant concentration, $\hat{y}_i$ is the predicted value of the air pollutant concentration, and L_2 is L_2 regularization, where λ is the regularization parameter, and W is the weight parameter of the network. The loss function distributes the calculated error to all layers of the network through backpropagation and uses the stochastic gradient descent algorithm to adjust the weights in the network until the network converges.

The EDSModel presented in this study was compared with other prediction models on the same dataset. Root mean square error (RMSE), mean absolute error (MAE), and correlation coefficient (R) were used as metrics to confirm the effectiveness of the proposed method. Experimental metrics were calculated by the following formulas:

$$RMSE(y, \hat{y}) = \sqrt{\frac{\sum_{i=1}^{T}(y_i - \hat{y}_i)^2}{T}} \tag{9}$$

$$MAE(y, \hat{y}) = \frac{1}{T}\sum_{i=1}^{T}|y_i - \hat{y}_i| \tag{10}$$

$$R(y, \hat{y}) = \frac{Cov(y, \hat{y})}{\sqrt{Var[y] * Var[\hat{y}]}} \tag{11}$$

where y_i is the observed value, $\hat{y}_i$ denotes the predicted value, T is the test set size, $Cov(y, \hat{y})$ is the covariance of y and $\hat{y}$, and $Var[y]$ and $Var[\hat{y}]$ represent the variance of y and $\hat{y}$, respectively.

4.2.3 Training

The hyperparameters in our EDSModel are determined during the training process, that is, the best performance model is selected on the validation set through the RMSE. We manually specify the hyperparameter ranges: learning rate {0.01, 0.005, 0.003, 0.001}, dropout rate {0.0, 0.1, 0.2, 0.3, 0.4, 0.5}, regularization parameter {0.1, 0.01, 0.001, 0.0001} and decay rate{0.99, 0.95, 0.90, 0.85}. For different datasets, we have found that the following setting work well: set the dropout to 0.5, decay rate to 0.99, regularization parameter to 0.0001 and learning rate to 0.001 for EDSModel. When using the comparison models, these settings still work well. We implemented EDSModel in Tensorflow, and train all models using the SGD optimizer with batch size 64.

4.2.4 Evaluation

The setting of hyperparameters in this study is based on the results of many experiments, leading to the final selection of the optimal set of hyperparameters. The validation set used in this study is closely related to the training stage, and after each epoch, the RMSE and MAE of the prediction model on the validation set are calculated. Therefore, the optimal model is selected based on the model error calculated on the validation set. The specific process is as follows: for each experiment, the number of epochs selected was 100. After training an epoch, we tested the trained model on the validation set. If the RMSE and MAE of the prediction model on the validation set became smaller, we updated and saved the model parameters. After many parameter

adjustments and experiments, when the prediction effect of the prediction model on the validation set was optimal, the training ended. Finally, get the prediction result by iterating all the samples in the test set.

4.3 Parameter Setting

To determine the optimal structure of EDSModel, we vary the number of layers of Encoder and compare the prediction results of EDSModels with different structures through experiments. The prediction performances of the EDSModel with different number of Encoder layers are shown in Table 2. When the number of Encoder layers is 1, the RMSE can reach a minimum of 22.3, and MAE can reach a minimum of 15.5. However, as the number of Encoder layers increases, the predictive performance of EDSModel increases very slowly or even over-fitting, but the training time rises rapidly. For example, when the number of Encoder network layers is 2, the RMSE value decreased from 22.3 to 22.1, but the training time increased by 7.5h. The experimental results suggest that it is necessary to balance prediction accuracy and training time in deciding the hyperparameters of the model. Therefore, the number of Encoder layers is 1, the prediction performance of EDSModel is best.

Table 2 Predictive performance of EDSModel with different layers [72-24h]

Model	RMSE	MAE	R	Run time
EDSModel - 1	22.3	15.5	0.74	8.3 h
EDSModel - 2	22.1	15.5	0.74	15.8 h
EDSModel - 3	23.9	16.3	0.70	22.6 h

In the experiment, dropout was used as a general trick to avoid model overfitting. According to historical experience and research results in the field of deep learning, the effect is obvious when the value of the training stage is 0.5. Therefore, in the training stage of different prediction models, the value of dropout is 0.5 for the hidden layer of the recurrent network, and the fully connected layer. In the verification and testing stage, for each model, the value of dropout is 1.0. After the experiments, the layer selection of the EDSModel is shown in Table 3, and the parameters used for model testing are shown in Table 3.

Table 3 Model parameter

Layer name	Output_size	Parameters	Values
RLSTM	128	layer nodes × number of layers	128×1
LSTM	128	layer nodes × number of layers	128×1
Full connected layer	256 128 1	layer nodes × number of layers	256×1 128×1 1×1
-	-	Batch_size	64
-	-	Dropout	0.5
-	-	Learning_rate	0.001
-	-	Epochs	100
-	-	λ	0.0001

4.4 Experimental Comparison

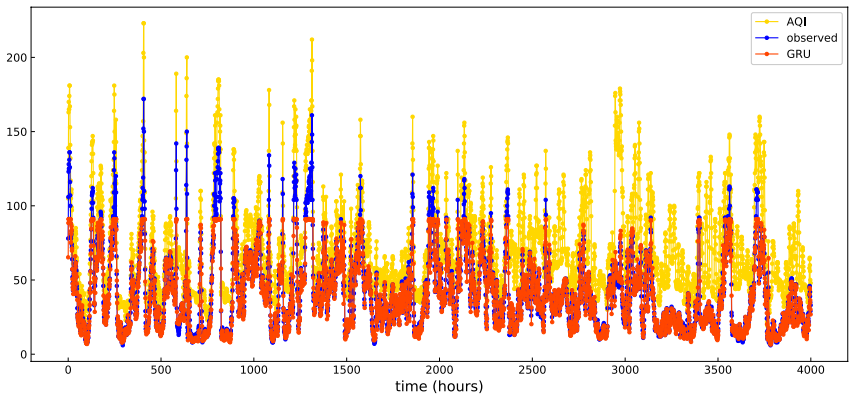
4.4.1 Prediction for the Next Hour

The existing pollutant concentration prediction methods are mostly used for the task of predicting the pollutant concentration in the next hour, that is, using the pollution data of the past r hours to predict the pollutant concentration in the next hour [1-4],[7,8],[15-18],[33-36],[38-44],[46],[48-56]. Therefore, for the pollutant concentration prediction task for the next hour, we compare EDSModel with the state-of-the-art models, including GRU [39], RNN [3],[51], LSTM [42],[44],[46], Bi-LSTM [43], and LSTM-Encoder-Decoder [54-56]. We use pollutant and meteorological data in the past 72 hours as input to the model to predict the pollutant concentration in the next hour. The experimental results are shown in Table 4.

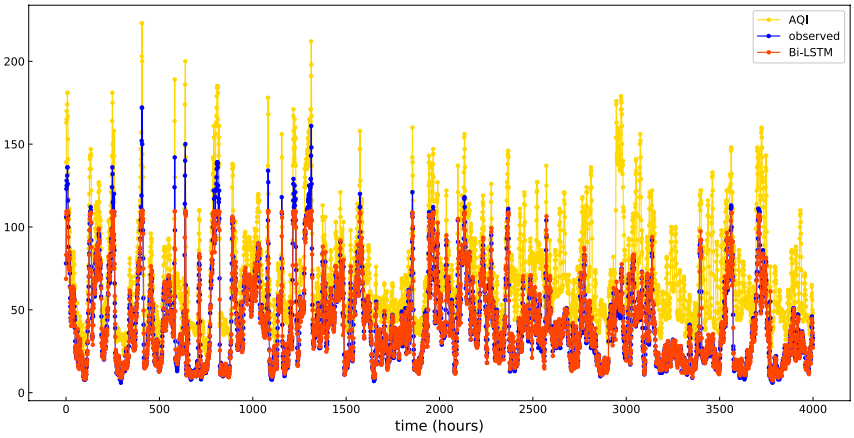
Table 4 The comparison of all models for the task [72-1h]

Method	RMSE	MAE	R
GRU [39]	7.9	4.4	0.96
RNN [3,51]	8.9	6.1	0.94
LSTM [42,44,46]	7.3	4.3	0.97
Bi-LSTM [43]	6.7	3.9	0.97
LSTM-Encoder-Decoder [54-56]	7.4	4.3	0.96
EDSModel	5.6	3.2	0.99

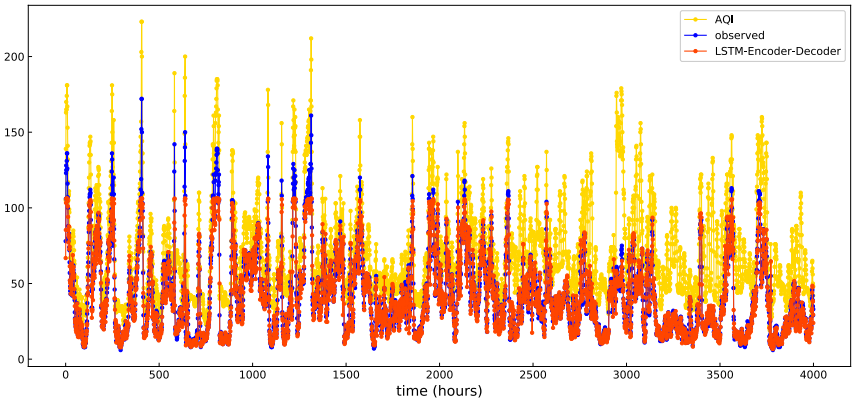
For the task of using pollutant and meteorological data in the past 72 hours to predict the pollutant concentration in the next hour, Fig. 6 shows the generalization ability of different models on the same test set. The length of Fig. 6's x-axis is 4000 hours, which means that 4000 consecutive hours were randomly selected in the test set to test the performance of the prediction model in this time period. We combine the prediction of pollutant with the change of AQI, and describe the location of mutation points more scientifically through AQI. According to the description of [58], when the AQI value fluctuates sharply, the mutation point appears. Therefore, we combine the test results with the mutation points to further verify the superiority of our EDSModel. The blue curve represents the observed value, the red curve represents the predicted value and the yellow curve represents the AQI value. Owing to space considerations in this study, Fig. 6 only shows the experimental results of the four state-of-the-art prediction models, representing the fitting trends of the GRU, Bi-LSTM, LSTM-Encoder-Decoder, EDSModel models were tested on the whole Shanghai test set.


AQI
observed
GRU
time (hours)

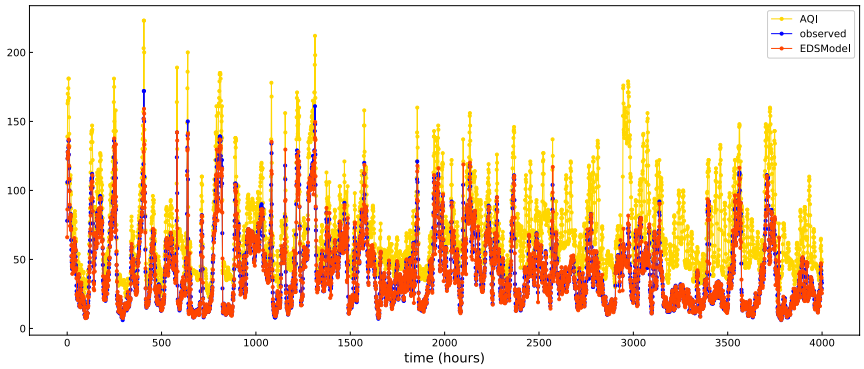
(a)


AQI
observed
Bi-LSTM
time (hours)

(b)


AQI
observed
LSTM-Encoder-Decoder
time (hours)

(c)


AQI
observed
EDSModel
time (hours)

(d)

Fig. 6. Fitting trends of the different models. (a)–(d) represent the fitting trends of GRU, Bi-LSTM, LSTM-Encoder-Decoder, EDSModel models.

To demonstrate the predictive performance of the EDSModel we chose, we compared it with the latest research results. We selected four prediction models, including the proposed EDSModel. Fig. 7 depicts the prediction performance of different prediction models on the test set. The x-axis represents the observed value of $PM_{2.5}$ and the y-axis represents the predicted value of $PM_{2.5}$. The black line indicates the $y = \hat{y}$ function, and the black dots indicate the degree of deviation between the observed and predicted values. In the dispersion comparison, when the concentration of $PM_{2.5}$ is greater than 100, the dispersion of GRU is the largest, and that of EDSModel is the smallest, meaning that the prediction performance is the best. When the values of $PM_{2.5}$ are between 0 and 100, the dispersion degree of EDSModel is still the smallest. Fig. 7 shows that the EDSModel predicted values are generally consistent with the observed values. In the correlation comparison, in the whole Shanghai test set, the correlation coefficients R of GRU, Bi-LSTM, LSTM-Encoder-Decoder, EDSModel are 0.96, 0.97, 0.96, and 0.99, respectively, which means that the correlation between predicted values and observed values of EDSModel is the largest. The R^2 value between the observed and predicted data indicated that 98% of the explained variance was captured by the EDSModel.

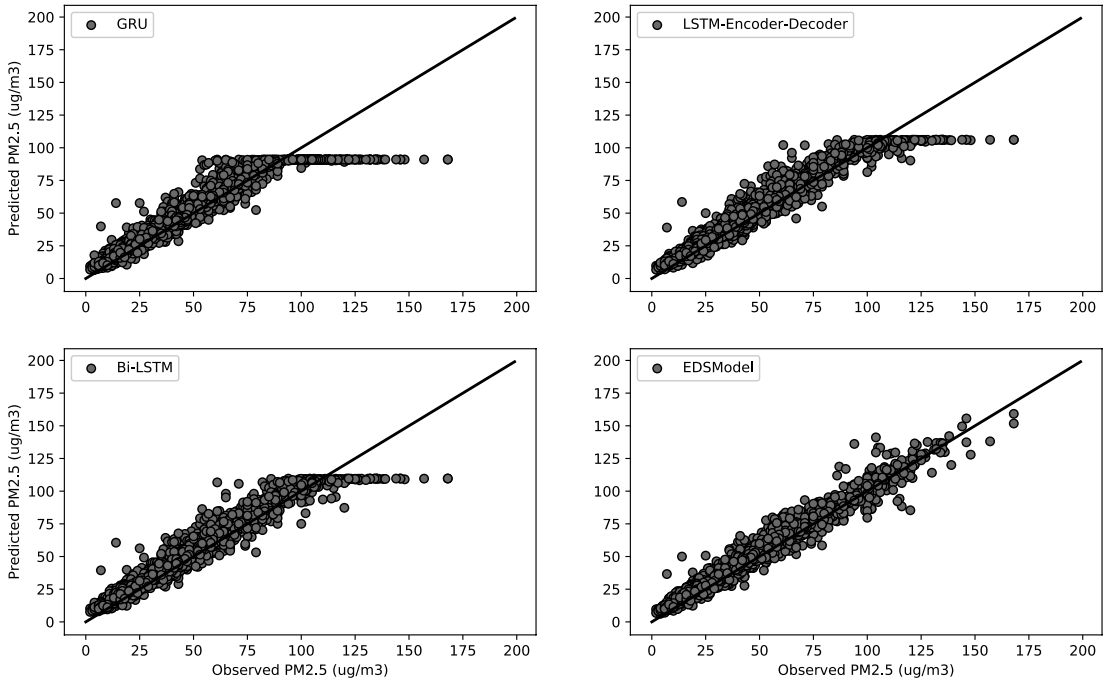


Fig. 7. Degree of fit between the observed and predicted values on the whole Shanghai test set

4.4.2 Prediction for the Time-series

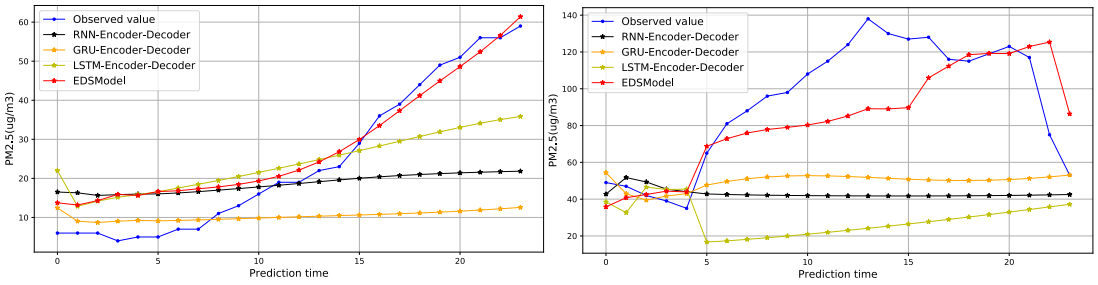
Research shows that long-term sequence feature extraction and time-series prediction are a difficult task [42-44],[57]. To show the advantages of EDSModel in extracting long-term sequence features and pollutant concentration prediction tasks, we use pollutant and meteorological data from the past 72 hours as input to the model to predict

pollutant concentration in the next 24 hours. Table 5 lists the average RMSE values, MAE values, and R values for each model on the same whole Shanghai test set.

Table 5 The comparison of all models for the task [72-24h]

Method	MAE	RMSE	R
CAMx [5]	-	37.5	0.69
CMAQ [10]	-	36.3	0.68
NAOPMS [23]	-	40.8	0.67
WRF-Chen [6]	-	43.5	0.45
SVM [14]	35.5	47.4	0.54
HMM [26]	35.7	47.8	0.52
SVR [25]	34.3	45.5	0.54
Random Forest [47]	36.8	47.7	0.52
XGBoost [45]	32.1	43.2	0.54
BP [28]	30.0	41.9	0.56
MLP [27]	31.5	41.8	0.55
RNN [3,51]	27.4	36.3	0.56
GRU [39]	26.5	36.3	0.66
LSTM [42,44,46]	24.1	36.3	0.69
Bi-LSTM [43]	23.7	36.1	0.68
RLSTM	22.9	36.1	0.69
RNN-Encoder-Decoder	21.3	32.2	0.69
GRU- Encoder-Decoder	19.1	29.0	0.70
LSTM- Encoder-Decoder [54-56]	18.9	25.6	0.70
EDSModel	**15.5**	**22.3**	**0.74**

For the task of using pollutant and meteorological data in the past 72 hours to predict the pollutant concentration in the next 24 hours, Fig. 8 shows the generalization ability of different models on the same whole Shanghai test set. y-axis indicates the value of $PM_{2.5}$ (μg/m3), and x-axis represents each prediction time. The blue curve represents the observed value and the black, orange, yellow, and red curves represent the predicted values of RNN-Encoder-Decoder, GRU-Encoder-Decoder, LSTM-Encoder-Decoder, and EDSModel, respectively. Due to page limit, we only list the test results for four random time periods in Fig. 8, the detailed numerical comparison has been presented in detail in Table 5. Our method of selecting samples for the four time periods is random selection, that is, randomly selecting four non-overlapping samples from the whole Shanghai test set.



(a) (b)

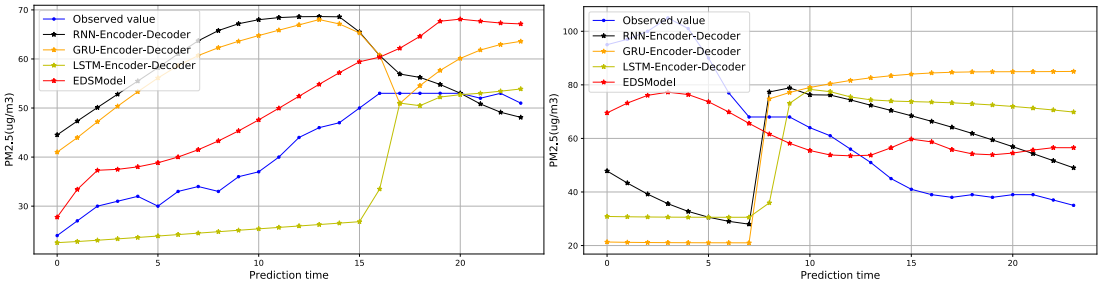


Fig. 8. Fitting trends of the different models, (a)-(d) represent the fitting trends of the RNN-Encoder-Decoder, GRU-Encoder-Decoder, LSTM-Encoder-Decoder, and EDSModel in four different time periods.

4.5 Predictive Model Generalization Ability

In order to verify the generalization ability and effectiveness of the EDSModel model proposed in this paper, we applied the trained EDSModel model to other cities in China for pollutant concentration prediction. We use Shanghai's monitoring data to train the prediction model proposed in this paper, and test the generalization ability of the model in nine cities in China.

Table 6 EDSModel generalization ability [72-24h]

City	RMSE	MAE	R
Shanghai	**22.3**	**15.5**	**0.74**
Nanjing	22.5	15.8	0.72
Hangzhou	22.6	16.1	0.72
Wuhan	22.3	15.9	0.73
Beijing	31.6	25.2	0.63
Shenyang	29.7	24.5	0.62
Harbin	27.1	24.1	0.64
Chengdu	25.4	20.3	0.71
Wulumuqi	27.4	22.8	0.68
Lasa	26.8	20.6	0.69

4.6 Trend Prediction

To further validate the proposed prediction model, we have extended the time length of pollutant concentration prediction, that is, we used the past 72 hours of pollutant and meteorological data as model inputs to predict trends in pollutant $PM_{2.5}$ concentrations over the next 48 hours. We compare EDSModel with the latest $PM_{2.5}$ prediction model, LSTM-Encoder-Decoder. Fig. 9 shows the predicted and observed changes in $PM_{2.5}$ over the next 48 hours (Randomly select samples from two different time periods on the whole Shanghai test set).

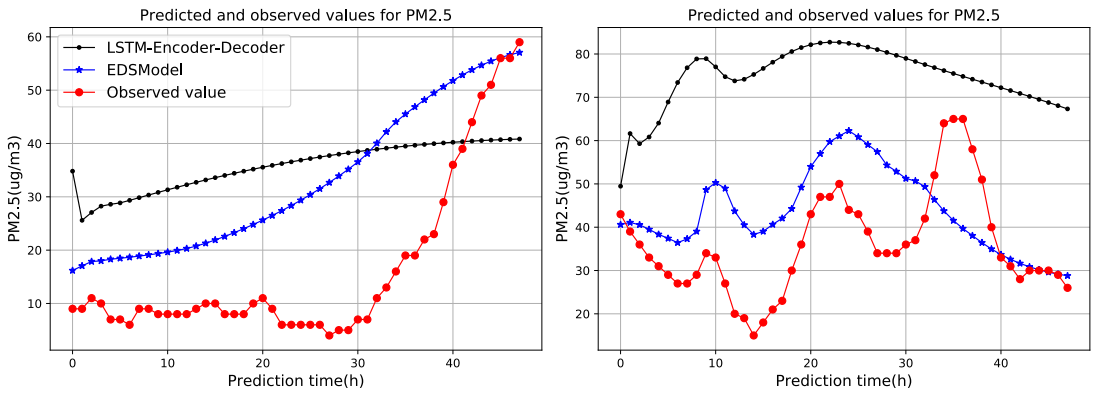


Fig. 9. Prediction of Shanghai pollutant concentration trends in the next 48 hours. The blue curve represents the predicted value of EDSModel, the black curve represents the predicted value of LSTM-Encoder-Decoder and the red curve represents the observed value.

5. Discussion

5.1 Comparison with Previous Prediction Models

5.1.1 Analysis of the Prediction Results for the Next Hour

Table 4 shows that, compared with the GRU and RNN methods, the LSTM, Bi-LSTM, LSTM-Encoder-Decoder, and EDSModel have better predictive results because all four can better handle long-term sequence dependency problems. Comparing the prediction results in Table 4 of RNN, GRU, and LSTM, the prediction accuracy of LSTM is higher than that of RNN and GRU, which proves that LSTM has better temporal feature ability to extract pollutant and meteorological data than RNN and GRU. Next, Comparing the prediction results in Table 4 of LSTM, Bi-LSTM, LSTM-Encoder-Decoder and EDSModel, the prediction accuracy of EDSModel is higher than that of LSTM, Bi-LSTM, and LSTM-Encoder-Decoder, which proves that deep EDSModel has better temporal feature ability to extract pollutant and meteorological data than LSTM, Bi-LSTM, and LSTM-Encoder-Decoder. Finally, by comparing the results in Table 4 of the LSTM-Encoder-Decoder and EDSModel experiments, it can be proved that RLSTM has better temporal feature extraction ability for long-term sequences than LSTM. The experimental results of the RCL-Learning model in Table 4 also confirm that the EDSModel is very effective for the prediction of $PM_{2.5}$. The RMSE optimal value is only 5.6, and the MAE optimal value is 3.2.

In this paper, 4000 consecutive test samples were randomly selected and presented in the experiment in the form of graph, as shown in Fig. 6 and Fig. 7. Therefore, our focus was on the fitting ability of the model to verify the supposition that EDSModel can better fit the mutation points. As shown in Fig. 6 and Fig. 7, when the $PM_{2.5}$ pollution source concentration is unstable, particularly when the concentration value is greater than 100, the prediction results of the comparison models could not follow the actual trend and showed a rather disordered pattern. This also reflects the fact that, in terms of the current $PM_{2.5}$ concentration prediction task, it is still difficult for the model to make

accurate predictions. Furthermore, the predictions and observations of the proposed EDSModel model are almost coincident and have a good fitting effect on the mutation of $PM_{2.5}$ concentration, such as the 46th hour, 165th hour, 288th hour, 444th hour, etc., as shown in Fig. 6.

Combining the fitting ability of each model in Fig. 6 and Fig. 7, we reach the following conclusions: (1) For the Fig. 6, we can get that the prediction performance of the EDSModel is better than the comparison models, and it is suitable for prediction tasks with sudden changes in pollutant concentration; (2) For the Fig. 7, we can get that compared with the comparison models, EDSModel can accurately predict high concentrations of $PM_{2.5}$, so that the predicted value and the observed value are highly consistent; (3) Combining the experimental results in Fig. 6 and Fig. 7, we can intuitively see that for mutation points, the $PM_{2.5}$ concentration is generally relatively high, and the number of mutation points is relatively small. This mainly reflects that in the general data set, the number of samples at mutation points is small, which leads to the problem of uneven data distribution. This phenomenon has caused the problem of insufficient learning of the predictive model, that is, it is difficult to learn the changing regularity of pollutant concentration under sudden changes. Therefore, this is also the reason why some models are difficult to fit in the case of sudden pollutant concentration.

Based on the above experimental results, our analysis result is that the EDSModel proposed in this paper tightly grasps the temporal characteristics of pollutants. In terms of data, we consider the impact of pollutants and meteorological factors in the pollutant concentration prediction task; In terms of the model, we utilize the RLSTM and LSTM as the temporal feature extractor, and make full use of the advantages of the two networks in feature extraction. Therefore, the characteristics of our prediction model are as follows: on the one hand, in large samples D_1 with small vibration amplitude of pollutant concentration, the changing regularity of pollutant concentration in historical data can be fully learned; on the other hand, in small samples D_2 with large fluctuations of pollutant concentration, we utilize the advantages of the EDSModel to learn the changing regularity of pollutant concentration, which can solve the problem that it is difficult to accurately predict the mutation of pollutants in the target city (training set=D_1+D_2). The ability of the EDSModel to predict $PM_{2.5}$ concentration is verified in this experiment.

5.1.2 Analysis of the Prediction Results for the Time-series

Table 5 shows the prediction errors of different models on the same test set. Table 5 shows that, compared with the four traditional models, five traditional machine learning methods and seven neural networks, the RNN-Encoder-Decoder, GRU-Encoder-Decoder, LSTM-Encoder-Decoder, and EDSModel models show better predictive results because Encoder-Decoder based on recurrent neural networks can better handle long-term sequence dependency problems. The RMSE reaches 22.3 to 32.2, MAE reaches 15.5 to 21.3, and R can reaches 0.69 to 0.74. Second, from Table 5, comparing

the prediction results of RNN-Encoder-Decoder, GRU-Encoder-Decoder, LSTM-Encoder-Decoder, and EDSModel, the prediction accuracy of EDSModel is higher than that of RNN-Encoder-Decoder, GRU-Encoder-Decoder and LSTM-Encoder-Decoder, which proves that EDSModel has better temporal feature extraction ability than RNN-Encoder-Decoder, GRU-Encoder-Decoder and LSTM-Encoder-Decoder. Its RMSE, MAE and R reach the optimal values of 22.3, 15.5 and 0.74, respectively. Third, as can be seen from Table 5, by comparing the results of LSTM-Encoder-Decoder and EDSModel, and comparing those of LSTM and RLSTM, it can be proved that RLSTM has better temporal feature extraction ability for long-term sequences than LSTM. However, using only the RLSTM model to extract the temporal features of complex pollutants and meteorological data, it is difficult to correlate the predicted values at different time. Therefore, this paper combines the advantages of RLSTM and LSTM, and proposes a new type of prediction framework, EDSModel. The experimental results of EDSModel in Table 5 also confirms that the combination of RLSTM and LSTM is very effective for the prediction of $PM_{2.5}$. The RMSE optimal value is only 22.3, and the MAE optimal value is 15.5.

In Fig. 8, (a)-(d) represent the fitting trends of the RNN-Encoder-Decoder, GRU-Encoder-Decoder, LSTM-Encoder-Decoder, and EDSModel models in four different time periods. Among them, the comparison model RNN-Encoder-Decoder, GRU-Encoder-Decoder, LSTM-Encoder-Decoder, and EDSModel models are trained on the same training set, and tested on the same test set. Combining the prediction results of Table 5 with the generalization ability of each model in Fig. 8, when the $PM_{2.5}$ pollution source concentration is unstable, the forecasting result of comparison models could not follow the real trend and showed a rather disordered pattern. This also indicates that it is still difficult in terms of $PM_{2.5}$ concentration. It shows that the predictions and observations of the proposed EDSModel are almost coincident and have a good fitting effect on the mutation of $PM_{2.5}$ concentration on a certain day, such as the figure (b), (c) and (d). This proves that EDSModel can better extract the temporal correlation features of complex pollutant concentration and meteorological data, solve the long-term dependence problem in pollutant prediction, and effectively cope with the sudden change of pollutant concentration. Overall, the performances of the RNN-Encoder-Decoder, GRU-Encoder-Decoder and LSTM-Encoder-Decoder are very stable and accurate, but the EDSModel proposed in this paper is even better. The predictive ability of EDSModel for $PM_{2.5}$ concentration is verified in this experiment.

5.2 Analysis Model Generalization Ability

The generalization of models on different data sets is difficult. It requires that the historical pollutant concentration and meteorological data be of high similarity. From the experimental results in Table 6, we can observe that for cities (Nanjing, Hangzhou, Wuhan) similar to Shanghai, EDSModel has strong generalization accuracy. For cities

with low similarity to Shanghai in terms of environment and distance (such as Beijing), the generalization accuracy of EDSModel is slightly reduced but is still satisfactory.

5.3 Analysis Model Trend Prediction

As shown in Fig. 9, LSTM-Encoder-Decoder curve fluctuation is small and stable, but it is difficult to predict the change trend of $PM_{2.5}$ concentration. The prediction accuracy of EDSModel is gradually decreasing with time in the next 48 hours, but it can accurately predict the future trend of pollutant concentration. From the figure we can see that the trend of the red observation curve and the blue prediction curve are consistent. The experiment verified that for the long-term prediction of pollutant concentration, the trend of pollutant concentration predicted by EDSModel has a strong correlation with the actual trend. Therefore, in the future pollutant prediction process, we can consider combining the trend of pollutant concentration predicted by the EDSModel with the state-of-the-art prediction methods, so as to more effectively improve the accuracy of pollutant prediction.

6. Conclusions

This paper studies how to improve pollutant concentration prediction, and proposes a new deep learning-based pollutant concentration prediction model, EDSModel. EDSModel is composed of an RLSTM-based Encoder and an LSTM-based Decoder. The experimental results show that the proposed EDSModel has a number of advantages. Compared with existing pollutant concentration prediction models, the RLSTM-based Encoder can better extract the temporal correlation features from the historical pollutant concentration and meteorological data. The LSTM-based Decoder correlates the hidden state of the Encoder output with the historical output of the Decoder to achieve a more accurate prediction of pollutant concentrations.

The experiments performed in this study demonstrated that, compared to traditional models, the proposed EDSModel yields higher-accuracy predictions by fully extracting data correlations, and overcomes problems such as long-term dependency. Therefore, the proposed EDSModel overcomes the weaknesses with traditional machine learning methods, single traditional networks and sequence network models based on RNN, GRU, and LSTM, and is valuable for practical applications. Compared with the traditional machine learning methods and single classical network, the EDSModel has been applied as one of the practical auxiliary models in the national urban air pollution monitoring and prediction tasks for many times, which shows good application effect and value. In addition, the distribution of pollutants has regional relevance, but this work does not consider the regional factor, which is left for our future work.

Acknowledgement

This work is funded by National Natural Science Foundation of China (61572326, 61802258, 61702333), Natural Science Foundation of Shanghai (18ZR1428300), the Shanghai Committee of Science and Technology (17070502800)

Reference

[1] Fong, I. H., Li, T., Fong, S., Wong, R. K., Tallón-Ballesteros, A. J., "Predicting concentration levels of air pollutants by transfer learning and recurrent neural network," Knowledge-Based Systems, vol. 192, 2020.

[2] Maleki, H., Sorooshian, A., Goudarzi, G., Baboli, Z., Birgani, Y. T., Rahmati, M., "Air pollution prediction by using an artificial neural network model," Clean Technologies and Environmental Policy, vol. 21, no. 6, pp. 1341-1352, 2019.

[3] Fan, J., Li, Q., Hou, J., Feng, X., Karimian, H., Lin, S., "A spatiotemporal prediction framework for air pollution based on deep RNN," ISPRS Annals of the Photogrammetry. Remote Sensing. Spatial Information Sciences, vol. 4, pp. 15-22, 2017.

[4] Feng, X., Li, Q., Zhu, Y., Hou, J., Jin, L., Wang, J., "Artificial neural networks forecasting of PM2.5 pollution using air mass trajectory based geographic model and wavelet transformation," Atmospheric Environment, vol. 107, pp. 118-128, 2015.

[5] Zhu, Y. Y., Gao, Y. X., Liu, B., Wang, X. Y., Zhu, L. L., Xu, R., Duan, X. L. "Concentration Characteristics and Assessment of Model-Predicted Results of PM2. 5 in the Beijing-Tianjin-Hebei Region in Autumn and Winter," Huan Jing ke Xue= Huanjing Kexue, vol. 40, no. 12, pp. 5191-5201, 2019.

[6] Saide, P. E., Carmichael, G. R., Spak, S. N., Gallardo, L., Osses, A. E., Mena-Carrasco, M. A., Pagowski, M., "Forecasting urban PM10 and PM2. 5 pollution episodes in very stable nocturnal conditions and complex terrain using WRF–Chem CO tracer model," Atmospheric Environment, vol. 45, no. 16, pp. 2769-2780, 2011.

[7] Huang, C. J., Kuo, P. H., "A deep cnn-lstm model for particulate matter (PM2. 5) forecasting in smart cities," Sensors, vol. 18, no. 7, pp. 2220, 2018.

[8] Park, S., Kim, M., Kim, M., Namgung, H. G., Kim, K. T., Cho, K. H., Kwon, S. B., "Predicting PM10 concentration in Seoul metropolitan subway stations using artificial neural network (ANN)," Journal of hazardous materials, vol. 341, pp. 75-82, 2018.

[9] Lee, A., Szpiro, A., Kim, S. Y., Sheppard, L., "Impact of preferential sampling on exposure prediction and health effect inference in the context of air pollution epidemiology," Environmetrics, vol. 26, no. 4, pp. 255-267, 2015.

[10] Chen, J., Lu, J., Avise, J. C., DaMassa, J. A., Kleeman, M. J., Kaduwela, A. P., "Seasonal modeling of PM2. 5 in California's San Joaquin Valley," Atmospheric environment, vol. 92, pp. 182-190, 2014.

[11] Cordano, M., Frieze, I. H., "Pollution reduction preferences of US environmental managers: Applying Ajzen's theory of planned behavior," Academy of Management journal, vol. 43, no. 4, pp. 627-641, 2000.

[12] Tian, J., Chen, D., "A semi-empirical model for predicting hourly ground-level fine particulate matter (PM2.5) concentration in southern Ontario from satellite remote sensing and ground-based meteorological measurements," Remote Sensing of Environment, vol. 114, no. 2, pp. 221-229, 2010.

[13] Russell, A. G., McCue, K. F., Cass, G. R., "Mathematical modeling of the formation of nitrogen-containing air pollutants. 1. Evaluation of an Eulerian photochemical model," Environmental science & technology, vol. 22, no. 3, pp. 263-271, 1988.

[14] Suleiman, A., Tight, M. R., Quinn, A. D., "Applying machine learning methods in managing urban concentrations of traffic-related particulate matter (PM10 and PM2. 5)," Atmospheric Pollution Research, vol. 10, no. 1, pp. 134-144, 2019.

[15] Li, X., Peng, L., Yao, X., Cui, S., Hu, Y., You, C., et al., "Long short-term memory neural network for air pollutant concentration predictions: method development and evaluation," Environmental Pollution, vol. 231, pp. 997-1004, 2017.

[16] Hossain, M., Rekabdar, B., Louis, S. J., Dascalu, S., "Forecasting the weather of Nevada: A deep learning approach," in Neural Networks (IJCNN), 2015 International Joint Conference on. IEEE, Killarney, Ireland, pp. 1-6, 2015.

[17] Gu, K., Qiao, J., Li, X., "Highly efficient picture-based prediction of PM2. 5 concentration," IEEE Transactions on Industrial Electronics, vol. 66, no. 4, pp. 3176-3184, 2019.

[18] Elbayoumi, M., Ramli, N. A., Yusof, N. F. F. M., "Development and comparison of regression models and feedforward backpropagation neural network models to predict seasonal indoor PM2. 5–10 and PM2. 5 concentrations in naturally ventilated schools," Atmospheric Pollution Research, vol. 6, no. 6, pp. 1013-1023, 2015.

[19] Gers, F. A. , Schmidhuber, Jürgen, Cummins, F., "Learning to forget: continual prediction with lstm," Neural Computation,vol. 12,no. 10, pp. 2451-2471, 2000.

[20] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.

[21] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on LSTM recurrent neural network," IEEE Transactions on Smart Grid, 2017.

[22] M. Sundermeyer, H. Ney, and R. Schlüter, "From feedforward to recurrent LSTM neural networks for language modeling," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 23, no. 3, pp. 517-529, 2015.

[23] Wang, Z., Maeda, T., Hayashi, M., Hsiao, L. F., Liu, K. Y., "A nested air quality prediction modeling system for urban and regional scales: Application for high-ozone episode in Taiwan," Water, Air, and Soil Pollution, vol. 130, no. 1-4, pp. 391-396, 2001.

[24] Corani, G., Scanagatta, M., "Air pollution prediction via multi-label classification," Environmental Modelling & Software, vol. 80, pp. 259-264, 2016.

[25] Yang, W., Deng, M., Xu, F., Wang, H., "Prediction of hourly PM2. 5 using a space-time support vector regression model," Atmospheric Environment, vol. 181, pp. 12-19, 2018.

[26] Sun, W., Zhang, H., Palazoglu, A., Singh, A., Zhang, W., Liu, S., "Prediction of 24-hour-average PM2. 5 concentrations using a hidden Markov model with different emission distributions in Northern California," Science of the total environment, vol. 443, pp. 93-103, 2013.

[27] Feng, R., Gao, H., Luo, K., Fan, J. R., "Analysis and accurate prediction of ambient PM2. 5 in China using Multi-layer Perceptron," Atmospheric Environment, 2020.

[28] Chen, Y., An, J., "A novel prediction model of PM2. 5 mass concentration based on back propagation neural network algorithm," Journal of Intelligent & Fuzzy Systems, vol. 37, no. 3, pp. 3175-3183, 2019.

[29] Hochreiter, S., Schmidhuber, J, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735-1780, 1997.

[30] Kim, H. Y., Won, C. H., “Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models,” Expert Systems with Applications, vol. 103, pp. 25-37, 2018.

[31] Huang, P. Y., Liu, F., Shiang, S. R., Oh, J., Dyer, C., “Attention-based multimodal neural machine translation,” In Proceedings of the First Conference on Machine Translation, vol. 2, pp. 639-645, August 2016.

[32] Yi, J., Wen, Z., Tao, J., Ni, H., Liu, B., “Ctc regularized model adaptation for improving lstm rnn based multi-accent mandarin speech recognition,” Journal of Signal Processing Systems, vol. 90, no. 7, pp. 985-997, 2018.

[33] Bui, T. C., Le, V. D., Cha, S. K., “A Deep Learning Approach for Air Pollution Forecasting in South Korea Using Encoder-Decoder Networks & LSTM,” in arXiv preprint arXiv:1804.07891, 2018.

[34] Yan, L., Wu, Y., Yan, L., and Zhou, M, “Encoder-Decoder Model for Forecast of PM2. 5 Concentration per Hour,” In 2018 1st International Cognitive Cities Conference (IC3), IEEE, pp. 45-50. August 2018.

[35] Gangopadhyay, T., Tan, S. Y., Huang, G., and Sarkar, S, “Temporal Attention and Stacked LSTMs for Multivariate Time Series Prediction,” in 32nd Conference on Neural Information Processing Systems, 2018.

[36] Liu, B., Yan, S., Li, J., Qu, G., Li, Y., Lang, J., and Gu, R, “An Attention-Based Air Quality Forecasting Method,” In 2018 17th IEEE International Conference on Machine Learning and Applications, (ICMLA), pp. 728-733, December 2018.

[37] Yang, J., Hu, M., “Filling the missing data gaps of daily MODIS AOD using spatiotemporal interpolation,” Science of The Total Environment, vol. 633, pp. 677-683, 2018.

[38] Qin, D., Yu, J., Zou, G., Yong, R., Zhao, Q., Zhang, B., “A novel combined prediction scheme based on CNN and LSTM for urban PM 2.5 concentration,” IEEE Access, vol. 7, pp. 20050-20059, 2019.

[39] Becerra-Rico, J., Aceves-Fernández, M. A., Esquivel-Escalante, K., Pedraza-Ortega, J. C., “ Airborne particle pollution predictive model using Gated Recurrent Unit (GRU) deep neural networks,” Earth Science Informatics, pp. 1-14, 2020.

[40] V. Le, T. Bui, S. Cha., “Spatiotemporal Deep Learning Model for Citywide Air Pollution Interpolation and Prediction,” *2020 IEEE International Conference on Big Data and Smart Computing*, pp. 55-62, 2020.

[41] Xu, Z., Lv, Y. , “Att-ConvLSTM: PM 2.5 Prediction Model and Application,” In *The International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery* , pp. 30-40, 2019.

[42] Karim R, Rafi T H., “An Automated LSTM-based Air Pollutant Concentration Estimation of Dhaka City,” Bangladesh[J]. Int. J. Eng. & Inf. Sys, vol. 4, no. 8, pp. 88-101, 2020.

[43] Zhang, B., Zhang, H., Zhao, G., Lian, J., “Constructing a PM2. 5 concentration prediction model by combining auto-encoder with Bi-LSTM neural networks,” *Environmental Modelling & Software*, vol. 124, 2020.

[44] Qadeer, K., Rehman, W. U., Sheri, A. M., Park, I., Kim, H. K., Jeon, M., “A Long Short-Term Memory (LSTM) Network for Hourly Estimation of PM2. 5 Concentration in Two Cities of South Korea,” *Applied Sciences*, vol. 10, no. 11, 2020.

[45] Zamani Joharestani, M., Cao, C., Ni, X., Bashir, B., Talebiesfandarani, S., "PM2. 5 Prediction based on random forest, XGBoost, and deep learning using multisource remote sensing data," *Atmosphere*, vol. 10, no. 7, 2019.

[46] Zhao, J., Deng, F., Cai, Y., Chen, J., "Long short-term memory-Fully connected (LSTM-FC) neural network for PM2. 5 concentration prediction," *Chemosphere*, vol. 220, pp. 486-492, 2019.

[47] Li, X., Zhang, X., "Predicting ground-level PM2. 5 concentrations in the Beijing-Tianjin-Hebei region: A hybrid remote sensing and machine learning approach," *Environmental Pollution*, vol. 249, pp. 735-749, 2019.

[48] Kim, H. S., Park, I., Song, C. H., Lee, K., Yun, J. W., Kim, H. K., Han, K. M., "Development of a daily PM10 and PM2. 5 prediction system using a deep long short-term memory neural network model," *Atmos. Chem. Phys*, vol. 19, pp. 12935-12951, 2019.

[49] Wang, X., Wang, B., "Research on prediction of environmental aerosol and PM2. 5 based on artificial neural network," *Neural Computing and Applications*, vol. 31, no. 12, pp. 8217-8227, 2019.

[50] Masmoudi, S., Elghazel, H., Taieb, D., Yazar, O., Kallel, A., "A machine-learning framework for predicting multiple air pollutants' concentrations via multi-target regression and feature selection," *Science of The Total Environment*, vol. 715, 2020.

[51] Chang-Hoi, H., Park, I., Oh, H. R., Gim, H. J., Hur, S. K., Kim, J., Choi, D. R., "Development of a PM2. 5 prediction model using a recurrent neural network algorithm for the Seoul metropolitan area, Republic of Korea," *Atmospheric Environment*, 2020.

[52] Hládek, D., Staš, J., Ondáš, S., "Comparison of Recurrent Neural Networks for Slovak Punctuation Restoration," In *2019 10th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, pp. 95-100, 2019.

[53] Zhang, P., Xue, J., Lan, C., Zeng, W., Gao, Z., Zheng, N., "Eleatt-rnn: Adding attentiveness to neurons in recurrent neural networks," *IEEE Transactions on Image Processing*, vol. 29, pp. 1061-1073, 2019.

[54] Kristiani, E., Yang, C. T., Huang, C. Y., Lin, J. R., Nguyen, K. L. P., "PM2. 5 Forecasting Using LSTM Sequence to Sequence Model in Taichung City," In *Information Science and Applications*, pp. 497-507, 2020.

[55] Lyu, P., Chen, N., Mao, S., Li, M., "LSTM based encoder-decoder for short-term predictions of gas concentration using multi-sensor fusion," *Process Safety and Environmental Protection*, vol. 137, pp. 93-105, 2020.

[56] Du, S., Li, T., Horng, S. J., "Time series forecasting using sequence-to-sequence deep learning framework," In *2018 9th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP)*, pp. 171-176, 2018.

[57] Jin, X., Yang, N., Wang, X., Bai, Y., Su, T., Kong, J., "Integrated predictor based on decomposition mechanism for PM2. 5 long-term prediction," *Applied Sciences*, vol. 9, no. 21, pp. 4533, 2019.

[58] Yi, X., Zhang, J., Wang, Z., Li, T., Zheng, Y. , "Deep distributed fusion network for air quality prediction," In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 965-973, 2018.

[59] Yang, B., Sun, S., Li, J., Lin, X., Tian, Y., “Traffic flow prediction using LSTM with feature enhancement,” *Neurocomputing*, vol, 332, pp, 320-327, 2019.
[60] Yao K, Cohn T, Vylomova K, Duh K, Dyer C., “Depth-gated recurrent neural networks,” arXiv preprint arXiv:1508.03790. 2015.