

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

Bo Zhang, Guojian Zou<sup>#</sup>, 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<sub>10</sub> and SO<sub>2</sub>). 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](mailto:zhangbo@shnu.edu.cn), [1000459475@mail.shnu.edu.cn](mailto:1000459475@mail.shnu.edu.cn), [kingpeng21@163.com](mailto: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](mailto: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](mailto:qindm@3clear.com))

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

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

\* Corresponding authors. Email: [h.wang@ulster.ac.uk](mailto:h.wang@ulster.ac.uk).

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

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

63 For the above problems in the pollutant concentration prediction task, deep learning  
64 technology has brought new solutions and performance improvements. Deep learning  
65 technology includes a variety of neural network models. These neural network models  
66 have different functions. Some networks have the advantage of extracting time series  
67 features, and some have the advantages of extracting spatial features, etc. To date, deep  
68 learning models have proved to be the state of the art in spatiotemporal prediction tasks  
69 [1-4],[7,8],[15-18],[33-36],[38-44],[46],[48-56]. In particular, the concentration of air  
70 pollutants is continuous in the temporal dimension, and the concentration of pollutants  
71 changes dynamically with time. Therefore, we can use the advantages of neural networks  
72 in extracting time series features to improve the accuracy of pollutant concentration  
73 prediction task. Many existing studies on air pollutant prediction have shown that deep  
74 learning models have better performance than traditional methods, including the

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

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

92 Our task is to solve the common weakness, extract the long-term serial characteristics  
93 of pollutant concentration and meteorological data, and ultimately achieve more  
94 accurate prediction of pollutant concentration. In this paper, we present a novel deep  
95 learning based air pollutant prediction method. It is an Encoder-Decoder model, named  
96 EDSModel, where the Encoder uses Read-first Long Short-Term Memory (RLSTM)  
97 and the Decoder uses Long Short-Term Memory (LSTM). (1) LSTM is a type of  
98 recurrent neural network (RNN) [15] that is able to predict future values using past ones.  
99 LSTM has been shown to be well-suited for time series prediction, with better  
100 performance than RNNs which suffer the exploding- and vanishing-gradient problems  
101 [20-22]. We use LSTM as the Decoder to continuously predict the concentration of air  
102 pollutants over a period of time, based on information stored in the Encoder. (2) RLSTM  
103 is a new model proposed in this paper, which is based on LSTM. In order to fully extract  
104 the long-term sequence features of pollutant concentration and meteorological data, we  
105 need to improve the ability of the LSTM model to extract features in the data encoding  
106 stage. The gating units of traditional LSTMs are independent of each other, which may  
107 lead to problems of low correlation in the feature extraction process and insufficient  
108 feature extraction for long-term sequences. RLSTM improves the traditional LSTM  
109 gating units to make the control gates interrelated, thereby improving the ability to  
110 extract long-term sequence features (i.e. the semantic information of the time series data  
111 extracted by the neural network model). To extract the temporal correlation between  
112 pollutant concentration and meteorological data, we use RLSTM as an Encoder to  
113 extract long-term sequence features from input data.

114 The main contributions of this paper are as follows:

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

131        **2. Related work**

132        According to the characteristics of the prediction methods used in related studies, air  
133        pollutant concentration prediction can be fundamentally divided into two major research  
134        methods: deterministic and statistical approaches [1-6],[8-10],[14],[18],[24-28],  
135        [33,34],[36],[38-56].

136        The deterministic approach can be applied to a limited set of historical data. However,  
137        meteorological principles and statistical approaches are needed to simulate the process  
138        of real-time emission, diffusion, transformation, and removal of pollutants based on  
139        atmospheric physics and chemical reactions. The model structure of a deterministic  
140        prediction method is predefined based on certain theoretical assumptions and prior  
141        knowledge. There are several commonly used deterministic methods for air pollutant  
142        concentration prediction: three-dimensional (3D) chemistry transport models (CTMs),  
143        the WRFChem model, nested air quality prediction modeling system (NAQPMS), and  
144        the community multiscale air quality (CMAQ) model [5],[6],[10],[23].

145        The statistical approach does not assume a complex theoretical model. Compared with  
146        the deterministic approach, it calculates statistics from complex pollutant concentration  
147        data and makes predictions on the basis of the statistics, usually showing better  
148        predictive performance than the deterministic approach. According to the type of  
149        statistics used, there are two branches of the statistical approach, traditional machine  
150        learning methods, and new deep learning methods. Traditional machine learning  
151        methods include support vector machine [14], multi-label classifier based on Bayesian  
152        networks [24], support vector regression (SVR) [25], hidden Markov model (HMM)  
153        [26], multiple linear regression (MLR) [18], XGBoost approach [45], and others [47,50].  
154        In recent years, deep learning technology has excelled in dealing with regression

155 problems, and various neural network based deep learning models have also been  
156 applied to improve air pollution concentration prediction performance. Typical network  
157 models include multi-layer perceptron (MLP) [27], artificial neural network(ANN)  
158 [8],[49], back propagation neural network (BP) [28], RNN neural network [3],[51],  
159 LSTM neural network [15],[42-44],[46],[48], deep CNN-LSTM model [7],[38], Gated  
160 Recurrent Unit (GRU) [39], Convolutional Long Short-Term Memory (ConvLSTM)  
161 [40], and attention-based neural networks [41]. Since air pollutant emissions, diffusion,  
162 conversion, and removal are a dynamic process over time, RNN is perhaps a good choice  
163 as it can process the time series prediction problem and easily extract temporal features  
164 of pollutant concentrations. Therefore, in previous studies, RNN has been used to predict  
165 the concentration of air pollutants. Nevertheless, the RNN has two shortcomings: long-  
166 term dependencies in input sequences cannot be captured; and a longer interval of input,  
167 or the number of RNN layers being larger, may cause a vanishing gradient or exploding  
168 gradient problems.

169 To solve these problems with RNN, Hochreiter and Schmidhuber presented an LSTM  
170 neural network model in 1997 [29]. The gate units in LSTM can selectively store  
171 information in the input sequence to capture the correlation between the long-term  
172 sequence data while solving the vanishing gradient problem. In recent years, LSTM has  
173 been successfully applied to many time series prediction problems, such as forecasting  
174 the daily maximum price of stocks, machine translation, and speech recognition [30-32].  
175 Moreover, LSTM is also widely used in air pollution prediction tasks, that is, using  
176 historical monitoring data of the city to predict the pollutant concentration [15],[42-  
177 44],[46],[48]. However, these prediction methods based on the LSTM network mainly  
178 predict the pollutant concentration at a certain time in the future, and do not make full  
179 use of multiple features in the pollutant data. At present, the pollutant prediction method  
180 based on LSTM network faces two key problems. (1) The gate units within LSTM are  
181 independent, so LSTM does not efficiently extract the long-term sequence  
182 characteristics of the input data [53],[59,60]. (2) Existing LSTM pollutant prediction  
183 models can only predict pollutant concentration at one time, and cannot accurately  
184 predict the concentration of pollutants in the future, that is, it is difficult to correlate the  
185 predicted values at each time [42-44].

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

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

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

### 208 3. Improved Long Short-term Memory Network

209 Air pollutant prediction is a typical time-series prediction problem, i.e., predicting the  
 210 value of the following period based on the sequence data of a known period. Studies  
 211 show that LSTM has good performance in air pollutant prediction tasks [15],[19],[42-  
 212 44],[46],[48]. This paper introduces the Encoder-Decoder model for pollutant prediction.  
 213 LSTM is modified, resulting in RLSTM, and used as the Encoder. This model aims to  
 214 solve the problem of long-term dependence and realize time-series prediction.  
 215 Compared with LSTM, RLSTM can more effectively process time-series data, compress  
 216 and extract the information of the historical pollutant concentration and meteorological  
 217 data. It can extract valid information from this data, drive the model to learn the  
 218 distribution features of historical data and prevent the dispersion of important  
 219 information, and finally lead EDSModel to predict the pollutant concentration of the  
 220 following period accurately. The nomenclature used in this paper is shown in Table 1.

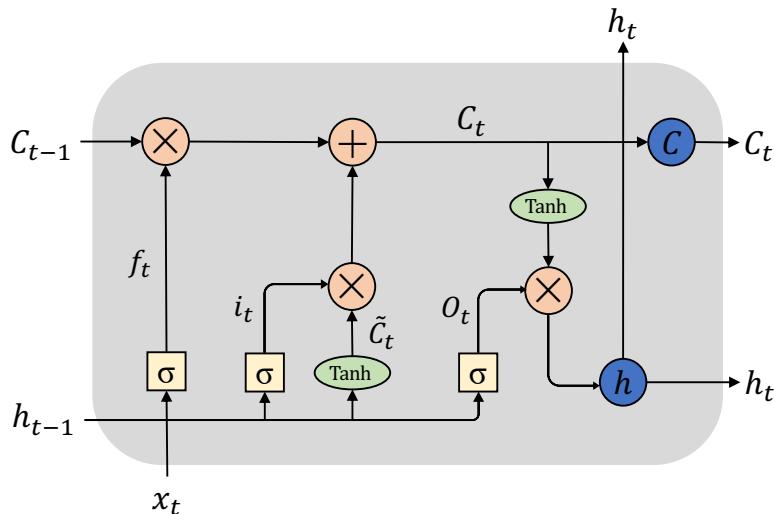
221

**Table 1** Nomenclature

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

222    3.1 Traditional LSTM

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



235    Fig. 1. Traditional LSTM cell.  $i$ ,  $f$ , and  $O$  represents input gate, forget gate and output gate,  
 236    respectively.  $x$  is the input feature,  $h$  is hidden state and  $C$  is cell memory state. ‘ $\times$ ’ and ‘ $+$ ’ represent  
 237    the multiplication and addition operations of the matrix, respectively.  $\sigma$  and  $Tanh$  are activation  
 238    functions.

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

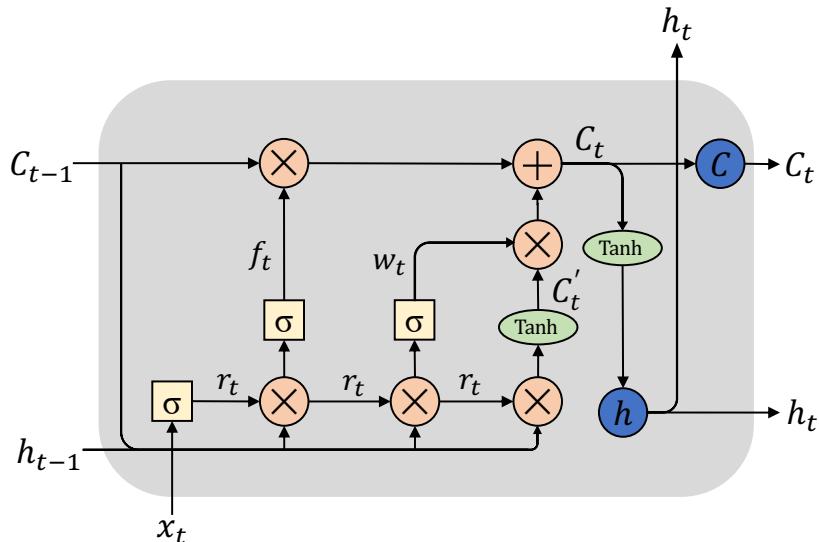
252 input gate  $f_t$  are two independent processes in LSTM. (3) Because the forget gate  $f_t$  and  
 253 input gate  $i_t$  are independent in the feature extraction process, the hidden feature  
 254 information  $h_t$  output by the output gate  $O_t$  may have problems such as feature  
 255 information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

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

### 259 3.2 RLSTM

#### 260 3.2.1 Feature Extraction Process

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



271  
 272 Fig. 2. An RLSTM cell is composed of gate units, memory unit, activation functions and matrix  
 273 operation.  $r$ ,  $f$ , and  $w$  represents read gate, forget gate and write gate, respectively.  $x$  is the input  
 274 feature,  $h$  is hidden state and  $C$  is cell memory state. ‘ $\times$ ’ and ‘ $+$ ’ represent the multiplication and  
 275 addition operations of the matrix, respectively.  $\sigma$  and  $Tanh$  are activation functions.

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

280 read gate uses the Sigmoid function as the activation function of the information filter  
 281 to constrain the range of feature information values. The filter method is similar to the  
 282 attention mechanism. That is, we use the sigmoid function to weight the input features  
 283  $[h_{t-1}, x_t, C_{t-1}]$ , a small weight indicates that the importance of the feature information  
 284 is less, and a large weight indicates that the feature is more important [53]. The weighting  
 285 process is completed through training of the neural network, and the weight ranges are  
 286 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  
 287 feature element 1 may be redundant information, and feature element 4 is important  
 288 information. This is shown in Equation (1):

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

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

294 Step 2. When updating the state information of the memory unit  $C_{t-1}$  at time  $t - 1$ ,  
 295 we need to select important input feature information, which has been completed in step  
 296 1; we also need to selectively forget the information in the memory unit  $C_{t-1}$ . Therefore,  
 297 we choose the Sigmoid function as the activation function of the forget gate. The  
 298 principle of the Sigmoid function here is the same as in the Step 1. Unlike LSTM,  
 299 RLSTM forgets the feature information with a small contribution in the memory unit

300  $C_{t-1}$  according to the output of the read gate  $r_t$  and the current input  $[h_{t-1}, x_t, C_{t-1}]$ .

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

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

303 The forget gate inputs the calculation result of the logistic regression function  $W_f \cdot (r_t *$   
 304  $[h_{t-1}, x_t, C_{t-1}]) + b_f$  into the Sigmoid function. The Sigmoid function assigns different  
 305 weight values to the feature elements in each dimension of the memory unit  $C_{t-1}$   
 306 according to the logistic regression function value, and the value is constrained between  
 307 [0.0,1.0]. By multiplying the output of the memory unit  $C_{t-1}$  and the forget gate  $f_t$ , a  
 308 part of the memory information is forgotten (if the weight value is 0.0, the corresponding  
 309 memory information is completely forgotten)

310 Step 3. After part of the state information of the memory unit is forgotten, RLSTM  
 311 needs to update the information of the memory unit according to the current read gate  $r_t$   
 312 and write gate  $w_t$ , that is, to perform the memory information write operation. The write  
 313 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  
 314 current time  $t$ , and selects some important feature information for updating the  
 315 information of the memory unit  $\hat{C}_t$  at the current time. Therefore, write gate  $w_t$  updates  
 316 the memory unit information based on the important features selected by the read gate  
 317  $r_t$ , thereby reducing redundant information residuals. This is shown in equations (4), (5)  
 318 and (6):

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

320                    $C'_t = \text{Tanh}(W_c \cdot (r_t * [h_{t-1}, x_t, C_{t-1}]) + b_c)$                    (5)

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

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

332       Therefore, the characteristics of RLSTM are summarised as follows:

333       1. It has three gates which are related to each other, and the function is different  
 334       from LSTM which RLSTM updates memory cell status information with low  
 335       redundancy; and

336       2. There are dependencies among its activation functions ( $\sigma$  and  $\text{Tanh}$ ).

### 337       3.2.2 Advantages of RLSTM

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

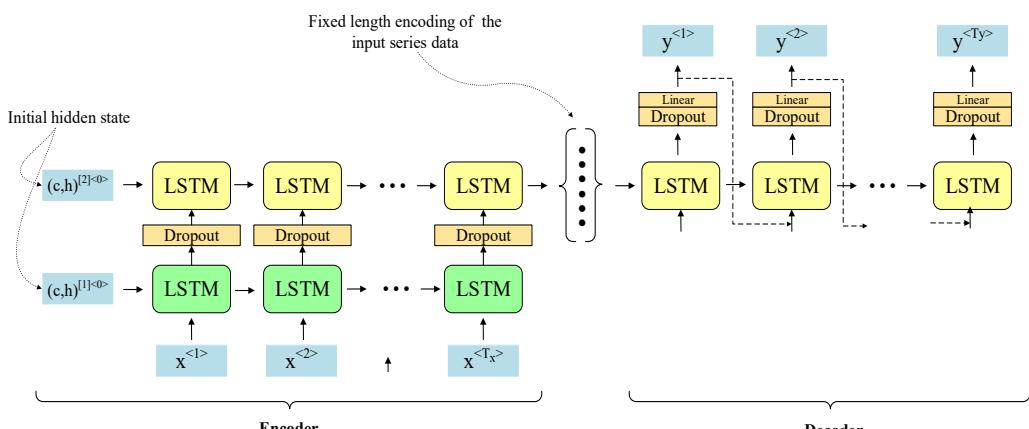
354       The equations of RLSTM are as follows:

355       
$$\begin{cases} r_t = \sigma(W_r \cdot [h_{t-1}, x_t, C_{t-1}] + b_r) \\ f_t = \sigma(W_f \cdot (r_t * [h_{t-1}, x_t, C_{t-1}]) + b_f) \\ w_t = \sigma(W_w \cdot (r_t * [h_{t-1}, x_t, C_{t-1}]) + b_w) \\ C'_t = \text{Tanh}(W_c \cdot (r_t * [h_{t-1}, x_t, C_{t-1}]) + b_c) \\ C_t = f_t * C_{t-1} + w_t * C'_t \\ h_t = \text{Tanh}(C_t) \end{cases} \quad (7)$$

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

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

380 *3.3 State-of-the-art Encoder-Decoder Model*



381 Fig. 3. Encoder-Decoder model. Using stacked LSTMs for encoding and one LSTM layer for  
 382 decoding.  
 383

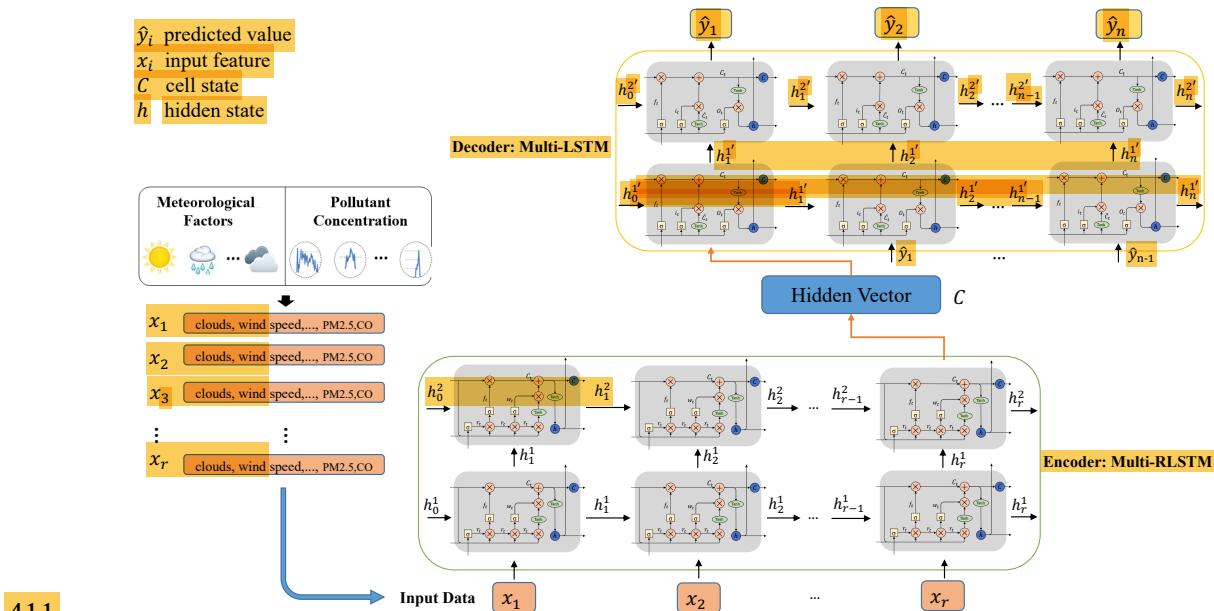
384 In computer science and related fields, the Encoder-Decoder model based on recurrent  
 385 neural network has been widely used in time series prediction tasks [33-36]. Recently,  
 386 the mainstream pollutant concentration prediction model is mainly based on LSTM

387 Encoder-Decoder, using one LSTM as the Encoder to extract the timing characteristics  
 388 of historical data, and then using another LSTM as the Decoder for time series prediction  
 389 [54-56]. The overall architecture of the model is shown in Fig. 3.

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

### 402 3.4 EDSModel for Air Pollutants

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



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

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

419 The EDSModel process is as follows:

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

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

#### 437 **4. Experimental results**

##### 438 *4.1 Data description*

439 The experiment used historical pollutant concentration and meteorological data from  
440 monitoring stations in 10 cities collected from May 13, 2014 to May 30, 2018 (Data  
441 sample and codes, URL: <http://github.com/zouguojian/data>). The experimental data in  
442 this paper is based on the city level, that is, the sample data of each city every hour is a  
443 one-dimensional feature vector, and the feature elements are composed of pollutants and  
444 meteorological factors. 10 cities are selected, Shanghai, Nanjing, Hangzhou, Wuhan,  
445 Beijing, Shenyang, Harbin, Chengdu, Wulumuqi, and Lasa, which have different  
446 economic development in China. The geographical location of these cities is scattered  
447 throughout the country, and the pollution environment of the cities varies. We selected  
448 16 pollutants and meteorological factors: AQI, PM<sub>2.5</sub>, PM<sub>10</sub>, SO<sub>2</sub>, NO<sub>2</sub>, O<sub>3</sub>, CO, Temp  
449 (temperature), Hum (humidity), air pressure, wind direction, wind speed, clouds,  
450 maximum temperature, minimum temperature and Conds (meteorological conditions).  
451 Fig. 5 shows the locations of all monitoring sites.



452

453 Fig. 5. Selected cities.

#### 454 4.2 Experimental Setup

##### 455 4.2.1 Datasets

456 In our experiment, we selected 70% of the data as the training set, 15% as validation  
 457 set, and the remaining 15% was used as the test set. The specific method of dividing  
 458 the data in this study is as follows: first, we divide the data set uniformly according to  
 459 a given window length  $L$  and a moving step size of  $S$ , and finally the total number of  
 460 samples obtained is  $N = ((D - D * 0.15) - L)/S$ ; then, we scramble the  $N$  samples,  
 461 select 82% of them as the training set and 18% as the validation set. In addition, 15%  
 462 of  $D$  is used as the test set, which means that we extract 15% of the data from the  
 463 original data set as the test set without disturbing it; finally, we define our division  
 464 method as a generalized random method. Among them, the window length  $L$  represents  
 465 the sum of the time sequence length of the input model and the target prediction  
 466 sequence length, and  $D$  is the size of the original data set. The missing values of the air  
 467 pollutant concentration and meteorological data set are filled by spatiotemporal  
 468 interpolation [37]. This paper attempted to predict the future  $n$  hour pollutant  
 469 concentration in the target city by using the pollutants and meteorological data in the  
 470 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  
 471 predicted value and  $x_i \in R^m$ , ( $m = 16$ ) represents the observed value.

##### 472 4.2.2 Related Definitions of EDSModel

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

$$476 \quad loss = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}} + L_2, (L_2 = \frac{\lambda}{2} \|W\|_2^2) \quad (8)$$

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

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

$$487 \quad RMSE(y, \hat{y}) = \sqrt{\frac{\sum_{i=1}^T (y_i - \hat{y}_i)^2}{T}} \quad (9)$$

$$488 \quad MAE(y, \hat{y}) = \frac{1}{T} \sum_{i=1}^T |y_i - \hat{y}_i| \quad (10)$$

$$489 \quad R(y, \hat{y}) = \frac{Cov(y, \hat{y})}{\sqrt{Var[y]*Var[\hat{y}]}} \quad (11)$$

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

#### 493 4.2.3 Training

494 The hyperparameters in our EDSModel are determined during the training process,  
495 that is, the best performance model is selected on the validation set through the RMSE.  
496 We manually specify the hyperparameter ranges: learning rate {0.01, 0.005, 0.003,  
497 0.001}, dropout rate {0.0, 0.1, 0.2, 0.3, 0.4, 0.5}, regularization parameter {0.1, 0.01,  
498 0.001, 0.0001} and decay rate {0.99, 0.95, 0.90, 0.85}. For different datasets, we have  
499 found that the following setting work well: set the dropout to 0.2, decay rate to 0.99,  
500 regularization parameter to 0.0001 and learning rate to 0.001 for EDSModel. When  
501 using the comparison models, these settings still work well. We implemented  
502 EDSModel in Tensorflow, and train all models using the SGD optimizer with batch  
503 size 128.

#### 504 4.2.4 Evaluation

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

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

517 *4.3 Parameter Setting*

518 To determine the optimal structure of EDSModel, we vary the number of layers of  
 519 Encoder and compare the prediction results of EDSModels with different structures  
 520 through experiments. The prediction performances of the EDSModel with different  
 521 number of Encoder layers are shown in Table 2. When the number of Encoder layers is  
 522 1, the RMSE can reach a minimum of 22.3, and MAE can reach a minimum of 15.5.  
 523 However, as the number of Encoder layers increases, the predictive performance of  
 524 EDSModel increases very slowly or even over-fitting, but the training time rises rapidly.  
 525 For example, when the number of Encoder network layers is 2, the RMSE value  
 526 decreased from 22.3 to 22.1, but the training time increased by 7.5h. The experimental  
 527 results suggest that it is necessary to balance prediction accuracy and training time in  
 528 deciding the hyperparameters of the model. Therefore, the number of Encoder layers is  
 529 1, the prediction performance of EDSModel is best.

530 **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

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

539 **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
	256		256×1
Full connected layer	128	layer nodes × number of layers	128×1
	1		1×1
-	-	Batch_size	64
-	-	Dropout	0.5
-	-	Learning_rate	0.001
-	-	Epochs	100
-	-	$\lambda$	0.009

540    *4.4 Experimental Comparison*541    ***4.4.1 Prediction for the Next Hour***

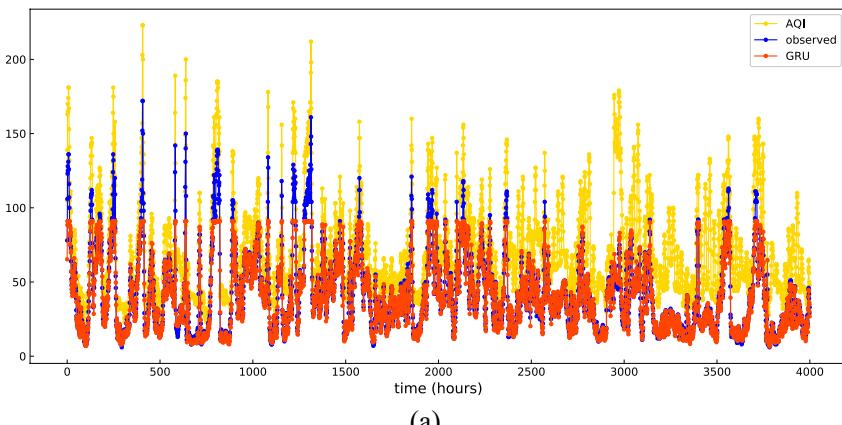
542    The existing pollutant concentration prediction methods are mostly used for the task  
 543    of predicting the pollutant concentration in the next hour, that is, using the pollution data  
 544    of the past  $r$  hours to predict the pollutant concentration in the next hour [1-4],[7,8],[15-  
 545    18],[33-36],[38-44],[46],[48-56]. Therefore, for the pollutant concentration prediction  
 546    task for the next hour, we compare EDSModel with the state-of-the-art models,  
 547    including GRU [39], RNN [3],[51], LSTM [42],[44],[46], Bi-LSTM [43], and LSTM-  
 548    Encoder-Decoder [54-56]. We use pollutant and meteorological data in the past 72 hours  
 549    as input to the model to predict the pollutant concentration in the next hour. The  
 550    experimental results are shown in Table 4.

551    **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
<b>EDSModel</b>	<b>5.6</b>	<b>3.2</b>	<b>0.99</b>

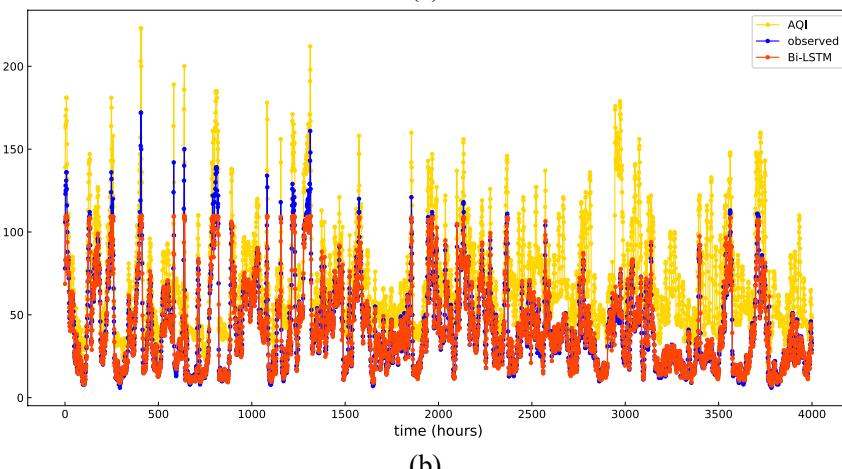
552    For the task of using pollutant and meteorological data in the past 72 hours to predict  
 553    the pollutant concentration in the next hour, Fig. 6 shows the generalization ability of  
 554    different models on the same test set. The length of Fig. 6's x-axis is 4000 hours, which  
 555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

566  
567



(a)

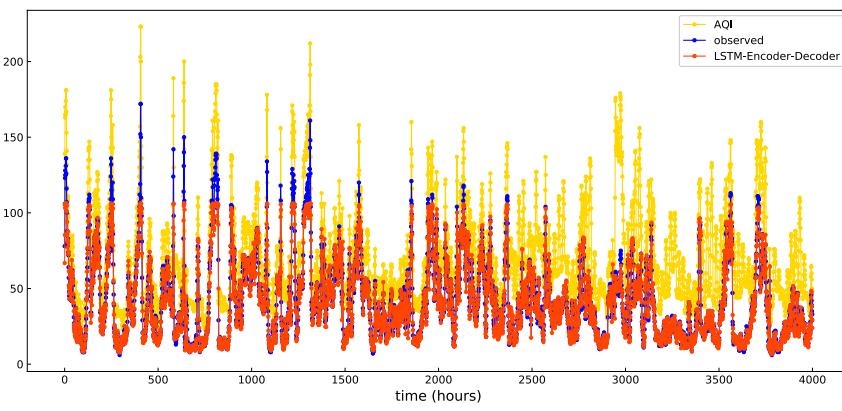
568  
569



(b)

570

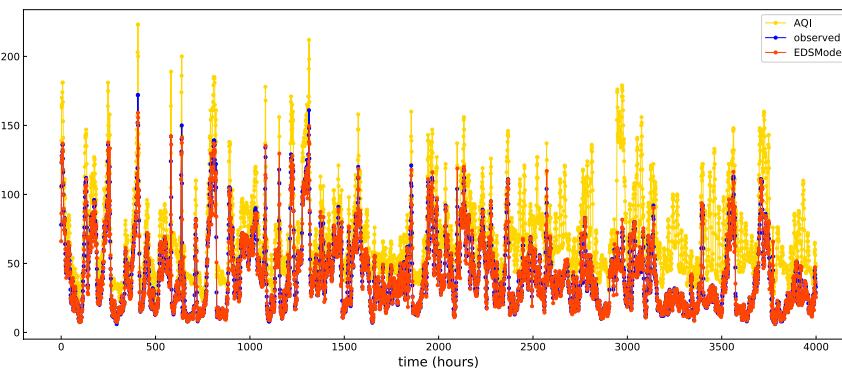
571



(c)

572

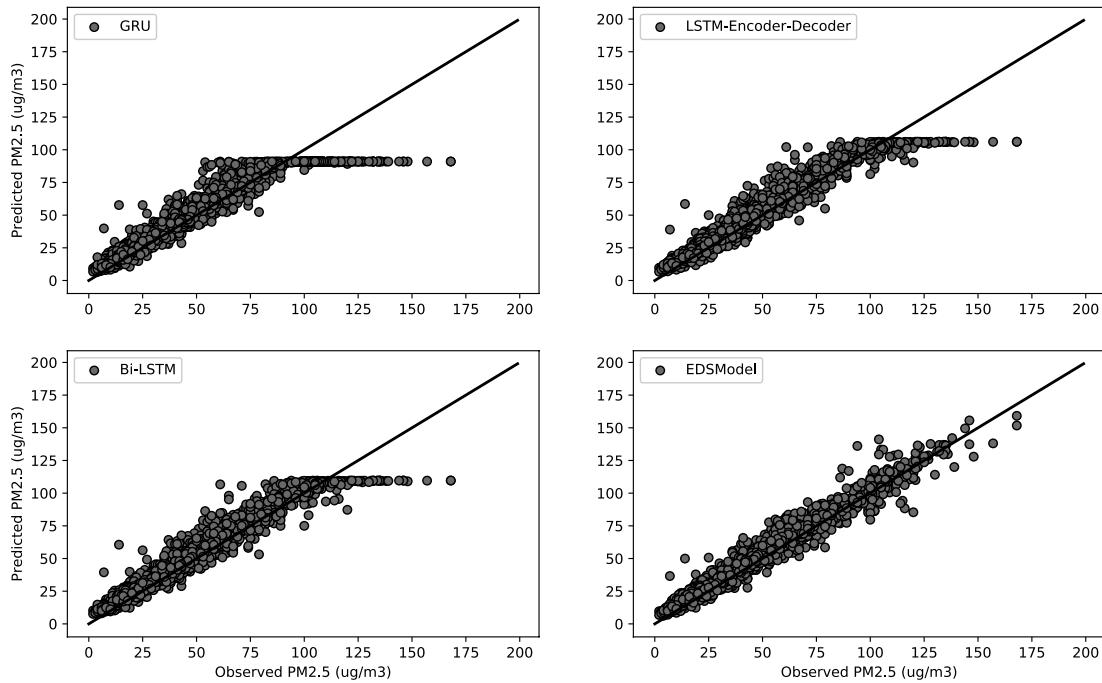
573



(d)

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

576 To demonstrate the predictive performance of the EDSModel we chose, we compared  
577 it with the latest research results. We selected four prediction models, including the  
578 proposed EDSModel. Fig. 7 depicts the prediction performance of different prediction  
579 models on the test set. The x-axis represents the observed value of PM<sub>2.5</sub> and the y-axis  
580 represents the predicted value of PM<sub>2.5</sub>. The black line indicates the  $y = \hat{y}$  function,  
581 and the black dots indicate the degree of deviation between the observed and predicted  
582 values. In the dispersion comparison, when the concentration of PM<sub>2.5</sub> is greater than  
583 100, the dispersion of GRU is the largest, and that of EDSModel is the smallest,  
584 meaning that the prediction performance is the best. When the values of PM<sub>2.5</sub> are  
585 between 0 and 100, the dispersion degree of EDSModel is still the smallest. Fig. 7  
586 shows that the EDSModel predicted values are generally consistent with the observed  
587 values. In the correlation comparison, in the whole Shanghai test set, the correlation  
588 coefficients R of GRU, Bi-LSTM, LSTM-Encoder-Decoder, EDSModel are 0.96, 0.97,  
589 0.976, and 0.99, respectively, which means that the correlation between predicted  
590 values and observed values of EDSModel is the largest. The R<sup>2</sup> value between the  
591 observed and predicted data indicated that 98% of the explained variance was captured  
592 by the EDSModel.



593  
594 Fig. 7. Degree of fit between the observed and predicted values on the whole Shanghai test set  
595 4.4.2 Prediction for the Time-series

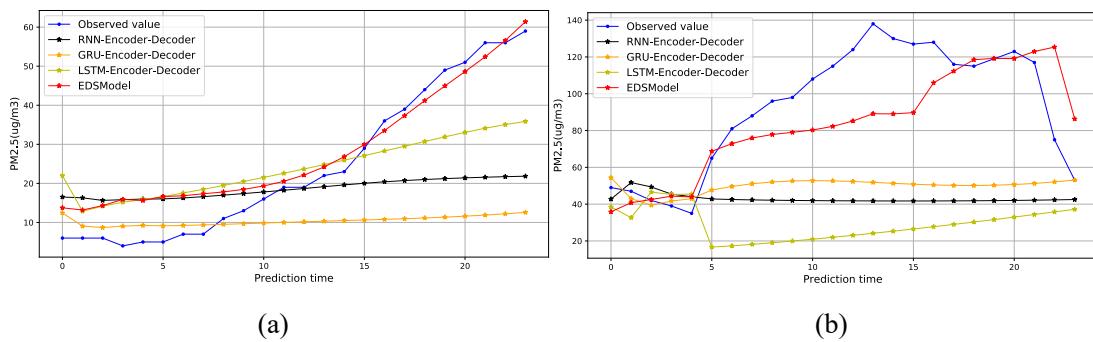
596 Research shows that long-term sequence feature extraction and time-series prediction  
597 are a difficult task [42-44],[57]. To show the advantages of EDSModel in extracting  
598 long-term sequence features and pollutant concentration prediction tasks, we use  
599 pollutant and meteorological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

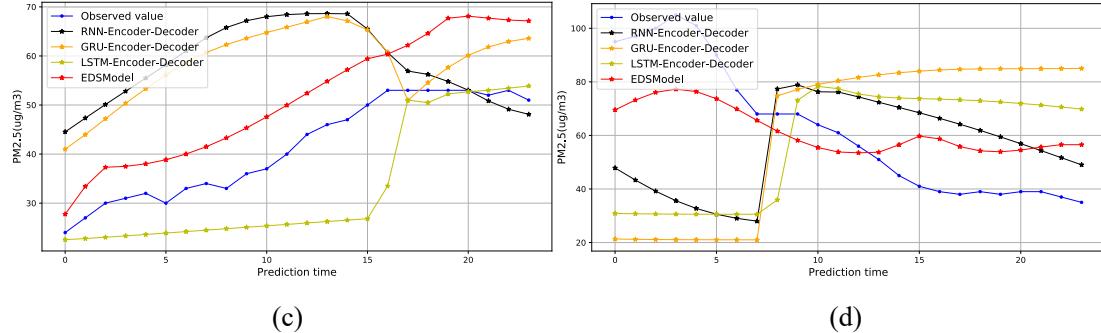
600 pollutant concentration in the next 24 hours. Table 5 lists the average RMSE values,  
 601 MAE values, and R values for each model on the same whole Shanghai test set.

602 **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
<b>EDSModel</b>	<b>15.5</b>	<b>22.3</b>	<b>0.74</b>

603 For the task of using pollutant and meteorological data in the past 72 hours to predict  
 604 the pollutant concentration in the next 24 hours, Fig. 8 shows the generalization ability  
 605 of different models on the same whole Shanghai test set. y-axis indicates the value of  
 606 PM<sub>2.5</sub>( $\mu\text{g}/\text{m}^3$ ), and x-axis represents each prediction time. The blue curve represents the  
 607 observed value and the black, orange, yellow, and red curves represent the predicted  
 608 values of RNN-Encoder-Decoder, GRU-Encoder-Decoder, LSTM-Encoder-Decoder,  
 609 and EDSModel, respectively. Due to page limit, we only list the test results for four  
 610 random time periods in Fig. 8, the detailed numerical comparison has been presented in  
 611 detail in Table 5. Our method of selecting samples for the four time periods is random  
 612 selection, that is, randomly selecting four non-overlapping samples from the whole  
 613 Shanghai test set.





616  
617 Fig. 8. Fitting trends of the different models, (a)-(d) represent the fitting trends of the RNN-Encoder-  
618 Decoder, GRU-Encoder-Decoder, LSTM-Encoder-Decoder, and EDSModel in four different time  
619 periods.  
620

#### 621 4.5 Predictive Model Generalization Ability

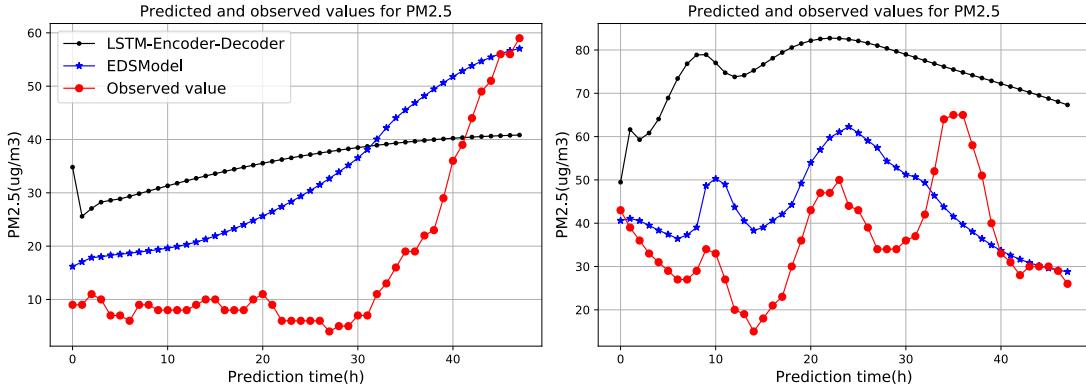
622 In order to verify the generalization ability and effectiveness of the EDSModel model  
623 proposed in this paper, we applied the trained EDSModel model to other cities in China  
624 for pollutant concentration prediction. We use Shanghai's monitoring data to train the  
625 prediction model proposed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

627 **Table 6** EDSModel generalization ability [72-24h]

City	RMSE	MAE	R
<b>Shanghai</b>	<b>22.3</b>	<b>15.5</b>	<b>0.74</b>
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

#### 628 4.6 Trend Prediction

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



636

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

## 640 5. Discussion

### 641 5.1 Comparison with Previous Prediction Models

#### 642 5.1.1 Analysis of the Prediction Results for the Next Hour

643 Table 4 shows that, compared with the GRU and RNN methods, the LSTM, Bi-  
 644 LSTM, LSTM-Encoder-Decoder, and EDSModel have better predictive results because  
 645 all four can better handle long-term sequence dependency problems. Comparing the  
 646 prediction results in Table 4 of RNN, GRU, and LSTM, the prediction accuracy of  
 647 LSTM is higher than that of RNN and GRU, which proves that LSTM has better  
 648 temporal feature ability to extract pollutant and meteorological data than RNN and  
 649 GRU. Next, Comparing the prediction results in Table 4 of LSTM, Bi-LSTM, LSTM-  
 650 Encoder-Decoder and EDSModel, the prediction accuracy of EDSModel is higher than  
 651 that of LSTM, Bi-LSTM, and LSTM-Encoder-Decoder, which proves that deep  
 652 EDSModel has better temporal feature ability to extract pollutant and meteorological  
 653 data than LSTM, Bi-LSTM, and LSTM-Encoder-Decoder. Finally, by comparing the  
 654 results in Table 4 of the LSTM-Encoder-Decoder and EDSModel experiments, it can  
 655 be proved that RLSTM has better temporal feature extraction ability for long-term  
 656 sequences than LSTM. The experimental results of the RCL-Learning model in Table  
 657 4 also confirm that the EDSModel is very effective for the prediction of PM<sub>2.5</sub>. The  
 658 RMSE optimal value is only 5.6, and the MAE optimal value is 3.2.

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

667 accurate predictions. Furthermore, the predictions and observations of the proposed  
668 EDSModel model are almost coincident and have a good fitting effect on the mutation  
669 of PM<sub>2.5</sub> concentration, such as the 46th hour, 165th hour, 288th hour, 444th hour, etc.,  
670 as shown in Fig. 6.

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

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

#### 699 5.1.2 Analysis of the Prediction Results for the Time-series

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

707 the prediction results of RNN-Encoder-Decoder, GRU-Encoder-Decoder, LSTM-  
708 Encoder-Decoder, and EDSModel, the prediction accuracy of EDSModel is higher than  
709 that of RNN-Encoder-Decoder, GRU-Encoder-Decoder and LSTM-Encoder-Decoder,  
710 which proves that EDSModel has better temporal feature extraction ability than RNN-  
711 Encoder-Decoder, GRU-Encoder-Decoder and LSTM-Encoder-Decoder. Its RMSE,  
712 MAE and R reach the optimal values of 22.3, 15.5 and 0.74, respectively. Third, as can  
713 be seen from Table 5, by comparing the results of LSTM-Encoder-Decoder and  
714 EDSModel, and comparing those of LSTM and RLSTM, it can be proved that RLSTM  
715 has better temporal feature extraction ability for long-term sequences than LSTM.  
716 However, using only the RLSTM model to extract the temporal features of complex  
717 pollutants and meteorological data, it is difficult to correlate the predicted values at  
718 different time. Therefore, this paper combines the advantages of RLSTM and LSTM,  
719 and proposes a new type of prediction framework, EDSModel. The experimental results  
720 of EDSModel in Table 5 also confirms that the combination of RLSTM and LSTM is  
721 very effective for the prediction of PM<sub>2.5</sub>. The RMSE optimal value is only 22.3, and  
722 the MAE optimal value is 15.5.

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

## 741 *5.2 Analysis Model Generalization Ability*

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

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

### 748 *5.3 Analysis Model Trend Prediction*

749 As shown in Fig. 9, LSTM-Encoder-Decoder curve fluctuation is small and stable, but  
750 it is difficult to predict the change trend of PM<sub>2.5</sub> concentration. The prediction accuracy  
751 of EDSModel is gradually decreasing with time in the next 48 hours, but it can accurately  
752 predict the future trend of pollutant concentration. From the figure we can see that the  
753 trend of the red observation curve and the blue prediction curve are consistent. The  
754 experiment verified that for the long-term prediction of pollutant concentration, the trend  
755 of pollutant concentration predicted by EDSModel has a strong correlation with the  
756 actual trend. Therefore, in the future pollutant prediction process, we can consider  
757 combining the trend of pollutant concentration predicted by the EDSModel with the  
758 state-of-the-art prediction methods, so as to more effectively improve the accuracy of  
759 pollutant prediction.

## 760 **6. Conclusions**

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

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

## 781 **Acknowledgement**

782 This work is funded by National Natural Science Foundation of China (61572326,  
783 61802258, 61702333), Natural Science Foundation of Shanghai (18ZR1428300), the  
784 Shanghai Committee of Science and Technology (17070502800)

785      **Reference**

- 786    [1] Fong, I. H., Li, T., Fong, S., Wong, R. K., Tallón-Ballesteros, A. J., "Predicting concentration  
787    levels of air pollutants by transfer learning and recurrent neural network," *Knowledge-Based  
788    Systems*, vol. 192, 2020.
- 789    [2] Maleki, H., Sorooshian, A., Goudarzi, G., Baboli, Z., Birgani, Y. T., Rahmati, M., "Air  
790    pollution prediction by using an artificial neural network model," *Clean Technologies and  
791    Environmental Policy*, vol. 21, no. 6, pp. 1341-1352, 2019.
- 792    [3] Fan, J., Li, Q., Hou, J., Feng, X., Karimian, H., Lin, S., "A spatiotemporal prediction framework  
793    for air pollution based on deep RNN," *ISPRS Annals of the Photogrammetry. Remote Sensing.  
794    Spatial Information Sciences*, vol. 4, pp. 15-22, 2017.
- 795    [4] Feng, X., Li, Q., Zhu, Y., Hou, J., Jin, L., Wang, J., "Artificial neural networks forecasting of  
796    PM2.5 pollution using air mass trajectory based geographic model and wavelet  
797    transformation," *Atmospheric Environment*, vol. 107, pp. 118-128, 2015.
- 798    [5] Zhu, Y. Y., Gao, Y. X., Liu, B., Wang, X. Y., Zhu, L. L., Xu, R., Duan, X. L. "Concentration  
799    Characteristics and Assessment of Model-Predicted Results of PM2. 5 in the Beijing-Tianjin-  
800    Hebei Region in Autumn and Winter," *Huan Jing ke Xue= Huanjing Kexue*, vol. 40, no. 12, pp.  
801    5191-5201, 2019.
- 802    [6] Saide, P. E., Carmichael, G. R., Spak, S. N., Gallardo, L., Osses, A. E., Mena-Carrasco, M. A.,  
803    Pagowski, M., "Forecasting urban PM10 and PM2. 5 pollution episodes in very stable nocturnal  
804    conditions and complex terrain using WRF-Chem CO tracer model," *Atmospheric  
805    Environment*, vol. 45, no. 16, pp. 2769-2780, 2011.
- 806    [7] Huang, C. J., Kuo, P. H., "A deep cnn-lstm model for particulate matter (PM2. 5) forecasting  
807    in smart cities," *Sensors*, vol. 18, no. 7, pp. 2220, 2018.
- 808    [8] Park, S., Kim, M., Kim, M., Namgung, H. G., Kim, K. T., Cho, K. H., Kwon, S. B., "Predicting  
809    PM10 concentration in Seoul metropolitan subway stations using artificial neural network  
810    (ANN)," *Journal of hazardous materials*, vol. 341, pp. 75-82, 2018.
- 811    [9] Lee, A., Szpiro, A., Kim, S. Y., Sheppard, L., "Impact of preferential sampling on exposure  
812    prediction and health effect inference in the context of air pollution epidemiology,"  
813    *Environmetrics*, vol. 26, no. 4, pp. 255-267, 2015.
- 814    [10] Chen, J., Lu, J., Avise, J. C., DaMassa, J. A., Kleeman, M. J., Kaduwela, A. P., "Seasonal  
815    modeling of PM2. 5 in California's San Joaquin Valley," *Atmospheric environment*, vol. 92, pp.  
816    182-190, 2014.
- 817    [11] Cordano, M., Frieze, I. H., "Pollution reduction preferences of US environmental managers:  
818    Applying Ajzen's theory of planned behavior," *Academy of Management journal*, vol. 43, no.  
819    4, pp. 627-641, 2000.
- 820    [12] Tian, J., Chen, D., "A semi-empirical model for predicting hourly ground-level fine particulate  
821    matter (PM2.5) concentration in southern Ontario from satellite remote sensing and ground-  
822    based meteorological measurements," *Remote Sensing of Environment*, vol. 114, no. 2, pp. 221-  
823    229, 2010.
- 824    [13] Russell, A. G., McCue, K. F., Cass, G. R., "Mathematical modeling of the formation of  
825    nitrogen-containing air pollutants. 1. Evaluation of an Eulerian photochemical  
826    model," *Environmental science & technology*, vol. 22, no. 3, pp. 263-271, 1988.

- 827 [14] Suleiman, A., Tight, M. R., Quinn, A. D., "Applying machine learning methods in managing  
828 urban concentrations of traffic-related particulate matter (PM10 and PM2. 5)," Atmospheric  
829 Pollution Research, vol. 10, no. 1, pp. 134-144, 2019.
- 830 [15] Li, X., Peng, L., Yao, X., Cui, S., Hu, Y., You, C., et al., "Long short-term memory neural  
831 network for air pollutant concentration predictions: method development and  
832 evaluation," Environmental Pollution, vol. 231, pp. 997-1004, 2017.
- 833 [16] Hossain, M., Rekabdar, B., Louis, S. J., Dascalu, S., "Forecasting the weather of Nevada: A  
834 deep learning approach," in Neural Networks (IJCNN), 2015 International Joint Conference on.  
835 IEEE, Killarney, Ireland, pp. 1-6, 2015.
- 836 [17] Gu, K., Qiao, J., Li, X., "Highly efficient picture-based prediction of PM2. 5 concentration,"  
837 IEEE Transactions on Industrial Electronics, vol. 66, no. 4, pp. 3176-3184, 2019.
- 838 [18] Elbayoumi, M., Ramli, N. A., Yusof, N. F. F. M., "Development and comparison of regression  
839 models and feedforward backpropagation neural network models to predict seasonal indoor  
840 PM2. 5-10 and PM2. 5 concentrations in naturally ventilated schools," Atmospheric Pollution  
841 Research, vol. 6, no. 6, pp. 1013-1023, 2015.
- 842 [19] Gers, F. A. , Schmidhuber, Jürgen, Cummins, F., "Learning to forget: continual prediction with  
843 lstm," Neural Computation,vol. 12,no. 10, pp. 2451-2471, 2000.
- 844 [20] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm:  
845 Human trajectory prediction in crowded spaces," in Proceedings of the IEEE conference on  
846 computer vision and pattern recognition, 2016.
- 847 [21] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load  
848 forecasting based on LSTM recurrent neural network," IEEE Transactions on Smart Grid, 2017.
- 849 [22] M. Sundermeyer, H. Ney, and R. Schlüter, "From feedforward to recurrent LSTM neural  
850 networks for language modeling," IEEE/ACM Transactions on Audio, Speech, and Language  
851 Processing, vol. 23, no. 3, pp. 517-529, 2015.
- 852 [23] Wang, Z., Maeda, T., Hayashi, M., Hsiao, L. F., Liu, K. Y., "A nested air quality prediction  
853 modeling system for urban and regional scales: Application for high-ozone episode in  
854 Taiwan," Water, Air, and Soil Pollution, vol. 130, no. 1-4, pp. 391-396, 2001.
- 855 [24] Corani, G., Scanagatta, M., "Air pollution prediction via multi-label classification,"  
856 Environmental Modelling & Software, vol. 80, pp. 259-264, 2016.
- 857 [25] Yang, W., Deng, M., Xu, F., Wang, H., "Prediction of hourly PM2. 5 using a space-time support  
858 vector regression model," Atmospheric Environment, vol. 181, pp. 12-19, 2018.
- 859 [26] Sun, W., Zhang, H., Palazoglu, A., Singh, A., Zhang, W., Liu, S., "Prediction of 24-hour-  
860 average PM2. 5 concentrations using a hidden Markov model with different emission  
861 distributions in Northern California," Science of the total environment, vol. 443, pp. 93-103,  
862 2013.
- 863 [27] Feng, R., Gao, H., Luo, K., Fan, J. R., "Analysis and accurate prediction of ambient PM2. 5 in  
864 China using Multi-layer Perceptron," Atmospheric Environment, 2020.
- 865 [28] Chen, Y., An, J., "A novel prediction model of PM2. 5 mass concentration based on back  
866 propagation neural network algorithm," Journal of Intelligent & Fuzzy Systems, vol. 37, no. 3,  
867 pp. 3175-3183, 2019.
- 868 [29] Hochreiter, S., Schmidhuber, J, "Long short-term memory," Neural computation, vol. 9, no. 8,  
869 pp. 1735-1780, 1997.

- 870 [30] Kim, H. Y., Won, C. H., "Forecasting the volatility of stock price index: A hybrid model  
871 integrating LSTM with multiple GARCH-type models," Expert Systems with Applications, vol.  
872 103, pp. 25-37, 2018.
- 873 [31] Huang, P. Y., Liu, F., Shiang, S. R., Oh, J., Dyer, C., "Attention-based multimodal neural  
874 machine translation," In Proceedings of the First Conference on Machine Translation, vol.  
875 2, pp. 639-645, August 2016.
- 876 [32] Yi, J., Wen, Z., Tao, J., Ni, H., Liu, B., "Ctc regularized model adaptation for improving lstm  
877 rnns based multi-accent mandarin speech recognition," Journal of Signal Processing Systems,  
878 vol. 90, no. 7, pp. 985-997, 2018.
- 879 [33] Bui, T. C., Le, V. D., Cha, S. K., "A Deep Learning Approach for Air Pollution Forecasting in  
880 South Korea Using Encoder-Decoder Networks & LSTM," in arXiv preprint  
881 arXiv:1804.07891, 2018.
- 882 [34] Yan, L., Wu, Y., Yan, L., and Zhou, M, "Encoder-Decoder Model for Forecast of PM2. 5  
883 Concentration per Hour," In 2018 1st International Cognitive Cities Conference (IC3),  
884 IEEE, pp. 45-50. August 2018.
- 885 [35] Gangopadhyay, T., Tan, S. Y., Huang, G., and Sarkar, S, "Temporal Attention and Stacked  
886 LSTMs for Multivariate Time Series Prediction," in 32nd Conference on Neural Information  
887 Processing Systems, 2018.
- 888 [36] Liu, B., Yan, S., Li, J., Qu, G., Li, Y., Lang, J., and Gu, R, "An Attention-Based Air Quality  
889 Forecasting Method," In 2018 17th IEEE International Conference on Machine Learning and  
890 Applications, (ICMLA), pp. 728-733, December 2018.
- 891 [37] Yang, J., Hu, M., "Filling the missing data gaps of daily MODIS AOD using spatiotemporal  
892 interpolation," Science of The Total Environment, vol. 633, pp. 677-683, 2018.
- 893 [38] Qin, D., Yu, J., Zou, G., Yong, R., Zhao, Q., Zhang, B., "A novel combined prediction scheme  
894 based on CNN and LSTM for urban PM 2.5 concentration," IEEE Access, vol. 7, pp. 20050-  
895 20059, 2019.
- 896 [39] Becerra-Rico, J., Aceves-Fernández, M. A., Esquivel-Escalante, K., Pedraza-Ortega, J. C., "  
897 Airborne particle pollution predictive model using Gated Recurrent Unit (GRU) deep neural  
898 networks," Earth Science Informatics, pp. 1-14, 2020.
- 899 [40] V. Le, T. Bui, S. Cha., "Spatiotemporal Deep Learning Model for Citywide Air Pollution  
900 Interpolation and Prediction," 2020 IEEE International Conference on Big Data and Smart  
901 Computing, pp. 55-62, 2020.
- 902 [41] Xu, Z., Lv, Y. , "Att-ConvLSTM: PM 2.5 Prediction Model and Application," In The  
903 International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery ,  
904 pp. 30-40, 2019.
- 905 [42] Karim R, Rafi T H., "An Automated LSTM-based Air Pollutant Concentration Estimation of  
906 Dhaka City," Bangladesh[J]. Int. J. Eng. & Inf. Sys, vol. 4, no. 8, pp. 88-101, 2020.
- 907 [43] Zhang, B., Zhang, H., Zhao, G., Lian, J., "Constructing a PM2. 5 concentration prediction model  
908 by combining auto-encoder with Bi-LSTM neural networks," Environmental Modelling &  
909 Software, vol. 124, 2020.
- 910 [44] Qadeer, K., Rehman, W. U., Sheri, A. M., Park, I., Kim, H. K., Jeon, M., "A Long Short-Term  
911 Memory (LSTM) Network for Hourly Estimation of PM2. 5 Concentration in Two Cities of  
912 South Korea," Applied Sciences, vol. 10, no. 11, 2020.

- 913 [45] Zamani Joharestani, M., Cao, C., Ni, X., Bashir, B., Talebiesfandarani, S., "PM2. 5 Prediction  
914 based on random forest, XGBoost, and deep learning using multisource remote sensing  
915 data," *Atmosphere*, vol. 10, no. 7, 2019.
- 916 [46] Zhao, J., Deng, F., Cai, Y., Chen, J., "Long short-term memory-Fully connected (LSTM-FC)  
917 neural network for PM2. 5 concentration prediction," *Chemosphere*, vol. 220, pp. 486-492,  
918 2019.
- 919 [47] Li, X., Zhang, X., "Predicting ground-level PM2. 5 concentrations in the Beijing-Tianjin-Hebei  
920 region: A hybrid remote sensing and machine learning approach," *Environmental  
921 Pollution*, vol. 249, pp. 735-749, 2019.
- 922 [48] Kim, H. S., Park, I., Song, C. H., Lee, K., Yun, J. W., Kim, H. K., Han, K. M., "Development  
923 of a daily PM10 and PM2. 5 prediction system using a deep long short-term memory neural  
924 network model," *Atmos. Chem. Phys*, vol. 19, pp. 12935-12951, 2019.
- 925 [49] Wang, X., Wang, B., "Research on prediction of environmental aerosol and PM2. 5 based on  
926 artificial neural network," *Neural Computing and Applications*, vol. 31, no. 12, pp. 8217-8227,  
927 2019.
- 928 [50] Masmoudi, S., Elghazel, H., Taieb, D., Yazar, O., Kallel, A., "A machine-learning framework  
929 for predicting multiple air pollutants' concentrations via multi-target regression and feature  
930 selection," *Science of The Total Environment*, vol. 715, 2020.
- 931 [51] Chang-Hoi, H., Park, I., Oh, H. R., Gim, H. J., Hur, S. K., Kim, J., Choi, D. R., "Development  
932 of a PM2. 5 prediction model using a recurrent neural network algorithm for the Seoul  
933 metropolitan area, Republic of Korea," *Atmospheric Environment*, 2020.
- 934 [52] Hládek, D., Staš, J., Ondáš, S., "Comparison of Recurrent Neural Networks for Slovak  
935 Punctuation Restoration," In *2019 10th IEEE International Conference on Cognitive  
936 Infocommunications (CogInfoCom)*, pp. 95-100, 2019.
- 937 [53] Zhang, P., Xue, J., Lan, C., Zeng, W., Gao, Z., Zheng, N., "Eleatt-rnn: Adding attentiveness to  
938 neurons in recurrent neural networks," *IEEE Transactions on Image Processing*, vol. 29, pp.  
939 1061-1073, 2019.
- 940 [54] Kristiani, E., Yang, C. T., Huang, C. Y., Lin, J. R., Nguyen, K. L. P., "PM2. 5 Forecasting Using  
941 LSTM Sequence to Sequence Model in Taichung City," In *Information Science and Applications*,  
942 pp. 497-507, 2020.
- 943 [55] Lyu, P., Chen, N., Mao, S., Li, M., "LSTM based encoder-decoder for short-term predictions  
944 of gas concentration using multi-sensor fusion," *Process Safety and Environmental  
945 Protection*, vol. 137, pp. 93-105, 2020.
- 946 [56] Du, S., Li, T., Horng, S. J., "Time series forecasting using sequence-to-sequence deep learning  
947 framework," In *2018 9th International Symposium on Parallel Architectures, Algorithms and  
948 Programming (PAAP)*, pp. 171-176, 2018.
- 949 [57] Jin, X., Yang, N., Wang, X., Bai, Y., Su, T., Kong, J., "Integrated predictor based on  
950 decomposition mechanism for PM2. 5 long-term prediction," *Applied Sciences*, vol. 9, no. 21,  
951 pp. 4533, 2019.
- 952 [58] Yi, X., Zhang, J., Wang, Z., Li, T., Zheng, Y., "Deep distributed fusion network for air quality  
953 prediction," In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge  
954 Discovery & Data Mining, pp. 965-973, 2018.

- 955 [59] Yang, B., Sun, S., Li, J., Lin, X., Tian, Y., "Traffic flow prediction using LSTM with feature  
956 enhancement," *Neurocomputing*, vol. 332, pp. 320-327, 2019.  
957 [60] Yao K, Cohn T, Vylomova K, Duh K, Dyer C., "Depth-gated recurrent neural networks," arXiv  
958 preprint arXiv:1508.03790. 2015.