Computer Modeling in Engineering & Sciences
DOI:10.32604/cmes.2020.012818
Article

Tech Science Press

Combining Trend-Based Loss with Neural Network for Air Quality Forecasting in Internet of Things

Weiwen Kong[1] and Baowei Wang[1,2,3,*]

[1]School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, 210044, China
[2]Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology, Nanjing University of Information Science and Technology, Nanjing, 210044, China
[3]Engineering Research Center of Digital Forensics, Ministry of Education, Nanjing, 210044, China
*Corresponding Author: Baowei Wang. Email: wang@nuist.edu.cn
Received: 13 July 2020; Accepted: 13 August 2020

Abstract: Internet of Things (IoT) is a network that connects things in a special union. It embeds a physical entity through an intelligent perception system to obtain information about the component at any time. It connects various objects. IoT has the ability of information transmission, information perception, and information processing. The air quality forecasting has always been an urgent problem, which affects people's quality of life seriously. So far, many air quality prediction algorithms have been proposed, which can be mainly classified into two categories. One is regression-based prediction, the other is deep learning-based prediction. Regression-based prediction is aimed to make use of the classical regression algorithm and the various supervised meteorological characteristics to regress the meteorological value. Deep learning methods usually use convolutional neural networks (CNN) or recurrent neural networks (RNN) to predict the meteorological value. As an excellent feature extractor, CNN has achieved good performance in many scenes. In the same way, as an efficient network for orderly data processing, RNN has also achieved good results. However, few or none of the above methods can meet the current accuracy requirements on prediction. Moreover, there is no way to pay attention to the trend monitoring of air quality data. For the sake of accurate results, this paper proposes a novel predicted-trend-based loss function (PTB), which is used to replace the loss function in RNN. At the same time, the trend of change and the predicted value are constrained to obtain more accurate prediction results of $PM_{2.5}$. In addition, this paper extends the model scenario to the prediction of the whole existing training data features. All the data on the next day of the model is mixed labels, which effectively realizes the prediction of all features. The experiments show that the loss function proposed in this paper is effective.

Keywords: Air quality forecasting; Internet of Things; recurrent neural network; predicted trend; loss function

1 Introduction

In recent years, the Internet of Things (IoT) [1] technology has been continuously improved, matured with the relevant researches, and widely used in many aspects [2]. The IoT conceptual model integrates devices. Based on this function, all node devices distributed in the grid are networked to transfer data [3]. Nowadays, air quality forecasting [4–6] has become a common but serious problem due to the demand for accurate meteorological data. So far, although there is a great quantity of real-time measurement methods [7–9], all of them will be seriously disturbed by external unexpected factors without exception. Moreover, these methods cannot predict future time data. For solving the prediction problem, a large number of machine learning algorithms were proposed and achieved good results [10–13].

Air quality forecasting is often seen as a regression problem due to its continuity [12,13]. Although some researchers regard it as a classification problem of multiple classes [11]. Without exception, they cannot get good prediction results and accuracy. Based on this, a large number of classical regression algorithms are applied to air quality prediction. As a classical regression algorithm, decision tree regression (DTR) [14] and support vector machine (SVM) [15] have achieved good results in air quality prediction. The heuristic method is used to partition the feature space. Each partition examines all the values of all the features in the current set one by one. According to the square error minimization criterion, the best strategy is selected as the segmentation point to realize data regression. In addition, gradient boost regression [16,17] is also applied to air quality forecasting. Gradient boost regression tree is a technique to learn mistakes. In essence, it is to gather ideas and integrate a bunch of poor learning algorithms for learning. Besides, other regression algorithms were also used on air quality forecasting, such as linear regression algorithm and local weighted regression.

Although the traditional regression algorithm can be applied to air quality prediction, these methods cannot meet the current accuracy requirements. For the sake of accuracy, a large number of deep learning models are used in air quality forecasting [18–21]. First, convolutional neural network (CNN) [22] has attracted more researchers' attention. Sahin et al. [19] used CNN to process the air quality data and achieved better performance than linear regression. An interesting result is that the concentrations of all pollutants are better predicted in winter than those in summer. Yi et al. [23] proposed a deep neural network-based approach to forecast air pollution data, which contains a spatial transformation subgroup and a distributed fusion network. Previously methods based on CNN were studied in greater depth and used larger data sets [24]. The former transforms sparse air quality data into consistent input to simulate pollution sources. The latter adopts the distributed structure of neural networks, integrates heterogeneous urban data, and captures the factors affecting air quality. Recently, more researchers have realized that air quality data is a time series data, which makes models with time series tasks get better prediction results on these data. Based on this, the recurrent neural network (RNN) [25] is more used in prediction. Tsai et al. [26] proposed a method to forecast $PM_{2.5}$ concentration using RNN with Long Short-Term Memory (LSTM) [27], which is the derivation of RNN. To deal with the missing value in series data, Fan et al. [28] proposed a spatiotemporal prediction framework based on deep recurrent neural networks (DRNN). The framework implements three different missing value fixing algorithms and integrates them into the deep neural network composed of the LSTM layer and fully connected layer. Moreover, to optimize the prediction model, Kim et al. [29] selected key input variables as a preprocessing step by the projection of the partial least squares (PLS) [30]. Athira et al. [31] proposed a framework where RNN, LSTM and GRU [32] were used for forecasting, based on the pollution and meteorological time series AirNet data.

However, these methods only pursue the prediction accuracy without exception, and ignore the changing trend of various pollution indicators. To solve this problem, this paper constructs a loss function which can monitor the trend of change. Also, this paper combines with the existing RNN model to realize the dual choice of prediction results and trends. To sum up, the main contributions of this paper are as follows:

(1) This article is based on the IoT technology to monitor the air quality data, through the use of IoT technology to complete the monitoring and management of air quality data. The monitoring system uses sensors, core control units, and wireless communication modules. They can help detect various air quality indicators to collect, analyze, and manage data.
(2) This paper proposes a novel loss function based on RNN model for forecasting the $PM_{2.5}$ value and all air quality data. This model not only predicts the accurate value effectively, but also monitors the changing trend of various indicators accurately.
(3) We verify the effectiveness and performance advantages of the proposed method through a large number of comparative experiments on the data in Beijing air quality data from 2018-01-01 to 2020-01-01.

2 Related Work

In this section some RNN models will be introduced, such as RNN, LSTM and GRU.

2.1 RNN

A traditional neural network only takes and processes the input information one by one. Moreover, there is no relevance between the near input information. However, some tasks need to handle the sequence information, where the traditional network cannot deal well with the previous input and the related later input. For the sake of some similar problems and dealing with sequence information better, RNN was proposed. The structure of RNN is shown as Fig. 1. In Fig. 1, X represents the value of the input layer while s represents the hidden layer. U is the weight matrix from the input layer to the hidden layer. o means the output layer; V is the weight matrix from the hidden layer to the output layer. The value s depends not only on the current input x, but also on the former hidden layer. W is the weight of the last value of the hidden layer as the input of this time. We can use the following equation to express the calculation method of the cyclic neural network:

$$\mathrm{o}_t = g\left(V\mathrm{s}_t\right) \tag{1}$$

$$\mathrm{s}_t = f\left(U\mathrm{x}_t + W\mathrm{s}_{t-1}\right) \tag{2}$$

Eq. (1) shows how to calculate the output layer, which is often a fully connected (FC) layer. where each node is connected with any other node. g is the activation function which is usually represented as *tanh* function. Eq. (2) is the calculation of the cyclic hidden layer. U is the weight matrix of input x, F is another activation function. If we repeat to bring Eq. (2) into Eq. (1), we will get:

$$\begin{aligned}
\mathrm{o}_t &= g\left(V\mathrm{s}_t\right) \\
&= Vf\left(U\mathrm{x}_t + Wf\left(U\mathrm{x}_{t-1} + W\mathrm{s}_{t-2}\right)\right) \\
&= Vf\left(U\mathrm{x}_t + Wf\left(U\mathrm{x}_{t-1} + Wf\left(U\mathrm{x}_{t-2} + W\mathrm{s}_{t-3}\right)\right)\right) \\
&= Vf\left(U\mathrm{x}_t + Wf\left(U\mathrm{x}_{t-1} + Wf\left(U\mathrm{x}_{t-2} + Wf\left(U\mathrm{x}_{t-3} + \cdots\right)\right)\right)\right)
\end{aligned} \tag{3}$$

It shows that the output value in t time, o_t is affected by the previous input, which is why RNN can deal with the former data sequence.

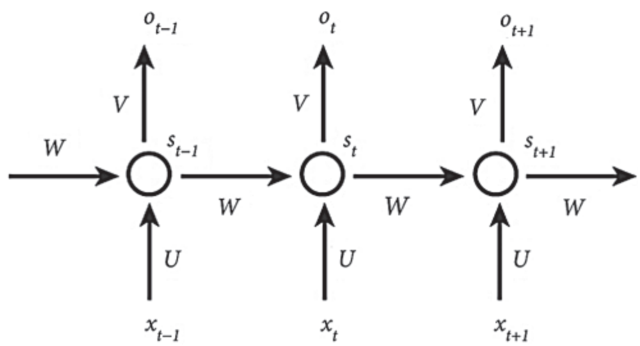


Figure 1: The structure of RNN

2.2 LSTM

Long short term memory (LSTM) is a kind of special RNN, which is an improvement of RNN on gradient processing, including gradient disappearance and gradient explosion. LSTM can perform better on a longer sequence than RNN. The structure of LSTM is shown in Fig. 2. Compared with RNN, LSTM has an extra structure for cell memory which indicates not only the output of the current cell need to be updated but also the state of the cell need to be considered. Specifically, the three gates are used for this function, which is forget gate, input gate and output gate, respectively. Forget gate can decide which information should be discarded. The information from the previous cell and the current cell is input into the sigmoid function. The output value is in [0,1]. This function can be used to control the data which is forgotten. The input gate is used to update the unit status which contains how to adjust the output into [0,1] and transform the input data by *tanh* function into [−1,1] to adjust the network, and then the *tanh* function output and the sigmoid function output are multiplied. The sigmoid output will determine which information is important and needs to be retained. The output gate can determine the value of the next hidden state, which contains the relevant input information previously. Also, hidden states can also be used for prediction. The overall process is as follows: First, the previous state and current state are transferred to the sigmoid function; Second, the new state is transferred by the *tanh* function and then the output will be multiplied to determine the information that the hidden state should carry; Finally, the hidden state outputs as the current cell, and the new state are transferred to the next state. The update rule in LSTM is shown as follows:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{4}$$

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \tag{5}$$

$$o_t = \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right) \tag{6}$$

$$C_t \approx \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right) \tag{7}$$

where f_t, i_t, o_t, C_t represent the state of forget gate, input gate, output data and the memory cell.

2.3 GRU

GRU is an effective variant of LSTM which is better but simpler than LSTM. The structure of GRU is shown in Fig. 3. In Fig. 3, there are only two doors: update gate and reset gate.

Update gate determines which information will be kept. Reset gate indicates how to process the previous information with the current information. The update guidelines in GRU is shown as follows.

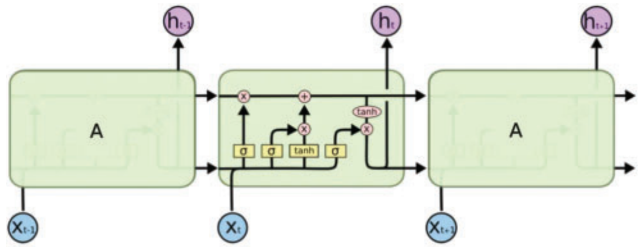


Figure 2: The structure of LSTM

$$z_t = \sigma \left(W_z * [x(t), h(t-1)] \right) \tag{8}$$

$$r_t = \sigma \left(W_r * [x(t), h(t-1)] \right) \tag{9}$$

$$\hat{h}(t) = \sigma \left(W_h * [x(t), (r_t * h(t-1))] \right) \tag{10}$$

$$h(t) = (1 - z_t) * h(t-1) + z_t * \hat{h}(t) \tag{11}$$

where σ is the activation function, $x(t)$ is the input, $h(t-1)$ is the previous output, W_z, W_r and W_h are the weights of the update gate, reset gate, and candidate output.

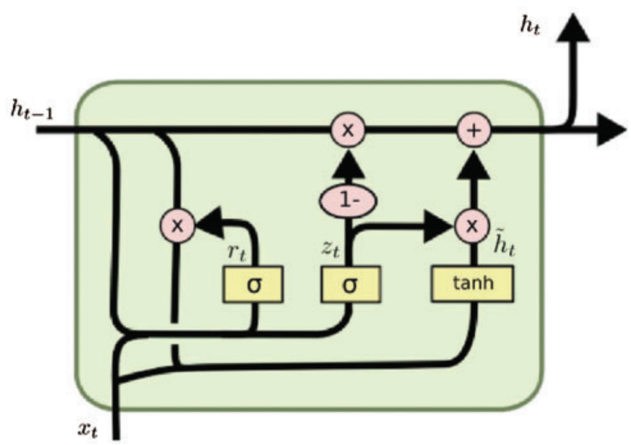


Figure 3: The structure of GRU

3 Air Quality Data Collection and Monitoring Scheme

For the need of real-time air quality information, this paper proposed a simple air quality monitoring system based on IoT to gather the pollution composition and analysis due to the poor prediction on air quality. At present, air quality monitoring technology is relatively backward. In order to supplement the inadequacy of the monitoring system, this article intends to set up monitoring nodes in qualified areas based on IoT.

The air quality index acquisition and data transmission module are constructed by a high-precision sensor module. The air composition index is analyzed and the current air quality status is obtained. In the experiment, some areas were set to nodes to collect the information in real time, obtain air composition information such as $PM_{2.5}$, PM_{10}, and SO_2, and analyze the real-time status.

The monitoring system needs to acquire and manage air composition and transmit to the database or platform through the network, which is shown in Fig. 4. In terms of function, air quality monitoring based on the IoT can be divided into three parts: data acquisition terminals, servers, and data processing platforms or databases. The data acquisition terminal is a ZigBee wireless sensor network. It mainly contains a coordinator node and several terminal nodes [33]. The terminal node includes sensors, core control units, wireless communication modules and embedded software systems for detecting various air composition indicators [35]. Its main function is to collect data. It collects data and uploads it to the server directly via wireless network. The server is equipped with a J2EE application server with the MySQL database, which is used to provide an interface for data access and save the location information and air quality data of monitoring points. The server can upload data to the background database and process platform. The platform in the background analyzes various air components and their impact on the air quality index.

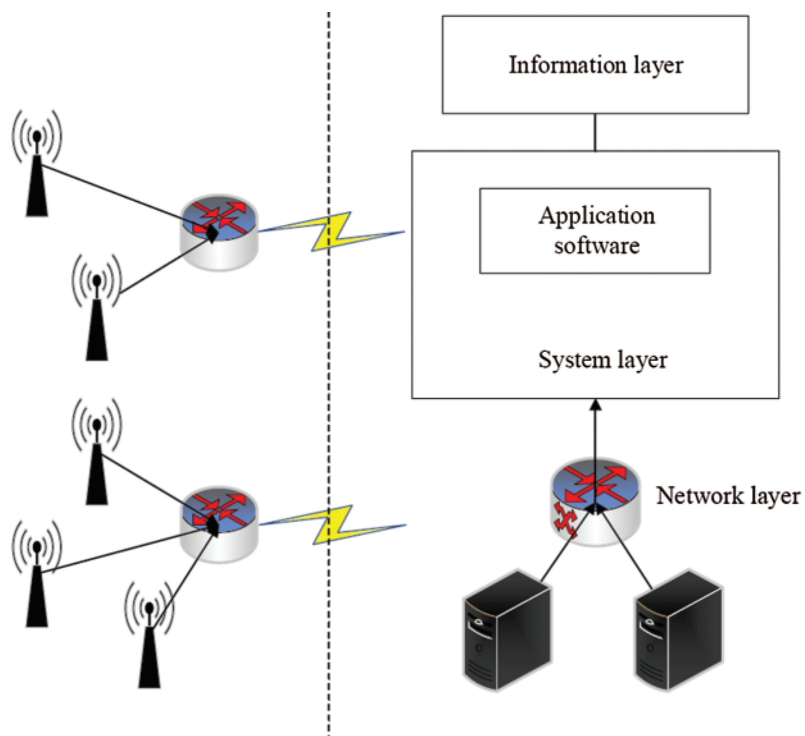


Figure 4: The structure of platform

3.1 Perception Technology

In a sensor network, each sensor node consists of sensors, microprocessors and communication units. Nodes form a sensor network through a communication network and work together to perceive and collect accurate information about the environment or objects. Currently, the wireless sensor network is the most widely used sensor network which is still increasing in usage [34]. The sensor is the main device for obtaining information in IoT. It uses measures to convert air quality data into electrical signals and then processed by specific signal processing equipment. Common sensors include temperature, pressure, photoelectric sensors, etc. The framework of the wireless sensor networks is shown in Fig. 5.

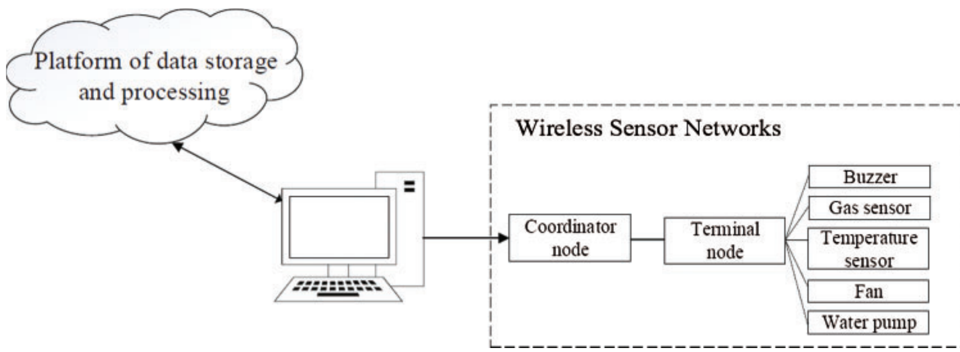


Figure 5: Air quality data monitoring framework based on IoT

3.2 Information Fusion Technology

Information fusion technology can carry out comprehensive analysis, collect various perception information and process this information, which helps the sensor network achieve many functions such as real-time monitoring, information management, real-time warning, intelligent decision-making. The monitoring data needs to be memorized, and the other node need to make a request to the master node to store this data. The master node selects the slave node and returns the storage location to the node requesting storage, and then the requesting storage node can directly send the data to the assigned slave node for storage. Through wireless sensor nodes to monitor the environment, multiple wireless sensor nodes use wireless routing nodes and base stations to form a wireless sensor network to analyze and display data, realizing real-time monitoring of air quality.

4 Proposed Method

In this section, we will introduce a novel predicted-trend-based loss function (PTB) to replace the traditional mean square error loss function on RNN models for predicting one demission $PM_{2.5}$ value data and multi-dimension air quality data.

4.1 PTB Loss Function for One Demission $PM_{2.5}$ Value Data

Among the common RNN models, MSE is the most common loss function, which is shown as follows:

$$MSE = \sum \left(y_{true} - y_{pred}\right)^2 \tag{12}$$

where y_{true} means the true value while y_{pred} means the predicted value. However, this single loss measurement method is difficult to adapt to the current requirements, especially in forecasting air quality data. When predicting air quality data, we should not only accurately predict the specific data at a certain time, but also predict the trend of data changes because of the continuance. An accurate trend of change is often more important than an accurate result. However, MSE loss function cannot meet this requirement. For the sake of continuity constraints, we propose a new loss function based on predicted value and true value, which is shown as follows:

$$loss = \sum \left(\Delta y_{true} - \Delta y_{pred}\right)^2 \tag{13}$$

where

$$\Delta y^i_{true} = \begin{cases} y^i_{true} - y^{i-1}_{true} & i > 1 \\ 0 & i = 1 \end{cases} \tag{14}$$

and

$$\Delta y^i_{pred} = \begin{cases} y^i_{pred} - y^{i-1}_{pred} & i > 1 \\ 0 & i = 1 \end{cases} \tag{15}$$

By this loss function, we can effectively control the rate of change on $PM_{2.5}$ value, so that the model can stably predict each trend on different features, so as to achieve more accurate prediction. The final loss function is the combination of MSE and the new loss function, which is called predicted-trend-based loss function and shown as follows:

$$loss = \sum \left(\left(y_{true} - y_{pred}\right)^2 + \alpha \left(\Delta y_{true} - \Delta y_{pred}\right)^2\right) \tag{16}$$

where α is the trade-off parameters. Based on this loss function, we can accurately predict the real data and the trend.

4.2 PTB Loss Function for Multi-Dimension Air Quality Data

In fact, $PM_{2.5}$ value and other air quality data are all needed to be predicted, such as PM_{10}, SO_2. The traditional method is to label each pollution data as a sample cell, and then predict each pollution data separately. This method not only sacrifices a great deal of time complexity, but also cannot judge the relationship between pollution features and the overall trend of multiple features. For the sake of this problem, based on Eq. (13), we propose a loss function for air quality feature prediction under more features. In the multi-dimension forecasting model, Y_{true} and Y_{pred} are represented for the true value and the predicted value. MSE is still the first part of the loss function, which is shown as follows:

$$MSE = \| Y_{true} - Y_{pred} \|_F^2 \tag{17}$$

where $||\cdot||_F$ means the Frobenius norm. In the same way, for the prediction of multiple features, we still need to consider the joint changing trend of each feature. For the sake of this problem, another loss function for multiple features is proposed, which is shown as follows:

$$loss = \|\Delta Y_{true} - \Delta Y_{pred}\|_F^2 \tag{18}$$

where

$$\Delta Y^i_{true} = \begin{cases} Y^i_{true} - Y^{i-1}_{true} & i > 1 \\ 0 & i = 1 \end{cases} \tag{19}$$

and

$$\Delta Y^i_{pred} = \begin{cases} Y^i_{pred} - Y^{i-1}_{pred} & i > 1 \\ 0 & i = 1 \end{cases} \tag{20}$$

In addition, to ensure the forecasting results are accurate rather than close enough, we take the l_1 norm as an additional constraint, and the final loss is shown as follows:

$$loss = \| Y_{true} - Y_{pred} \|_F^2 + \alpha \, \| \Delta Y_{true} - \Delta Y_{pred} \|_F^2 + \beta \, \| Y_{true} - Y_{pred} \|_1 \tag{21}$$

where α and β are the trade-off parameters.

5 Experiment

In this section, we will perform experiments on real air quality data to evaluate the proposed loss function based on RNN, LSTM and GRU. By comparing the traditional loss function with our PTB loss function, the prediction performance and effectiveness of the model are verified.

5.1 Dataset

The experience in this paper uses six cities air quality dataset which includes $PM_{2.5}$ value data and other feature such as date, time, PM_{10}. The data interval in the dataset is one hour and the dataset used for experiment is ranged from 2018-01-01 to 2020-01-01.

5.2 Setup

5.2.1 Error Measurement

We choose Root Mean Square Error (RMSE) as the error measurement while RMSE can better reflect the actual situation of the prediction error. The calculation formula is shown as follows:

$$RMSE_{(y_{pred}, y_{true})} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left(y_{pred}^i - y_{true}^i \right)^2} \tag{22}$$

where n is the number of samples, y_{true}^i is the real data, y_{pred}^i is the predicted data. Moreover, the other two evaluating index are used in our experiments, which is R_2 Square and false alarm rate (FAR). R_2 Square is shown as follows:

$$R_2 \ Square_{(y_{pred}, y_{true})} = 1 - \frac{\sum_{i=1}^{n} \left(y_{true}^i - y_{pred}^i \right)^2}{\sum_{i=1}^{n} \left(y_{true}^i - \bar{y} \right)^2} \tag{23}$$

which can evaluate the degree of change and accuracy of data, measuring the prediction quality of model. FAR is an index measuring error rate. In this paper, if $\frac{\left| y_{pred}^i - y_{true}^i \right|}{y_{true}^i} > 0.3$, the sample i will be a false sample, FAR is the proportion of false samples in the total number.

5.2.2 Experiment Setting

In this experiment, RNN, LSTM and GRU are selected as our baseline. Then we set the neurons of every layer as 200, where the number of layers is set as 3. Adam is chosen as the activation function; the dropout is 0.25 while the epoch is 200. Each result is taken for 10 rounds of the average value. When training, the weight matrices will be stored if the loss of the past epoch is greater than that of the current epoch. In addition, all the deep learning models use the early stop condition in the training process. If the loss of validation data does not change in 10 training epochs, the training step will be stopped.

Table 1: The RMSE value of different methods in six cities or regions

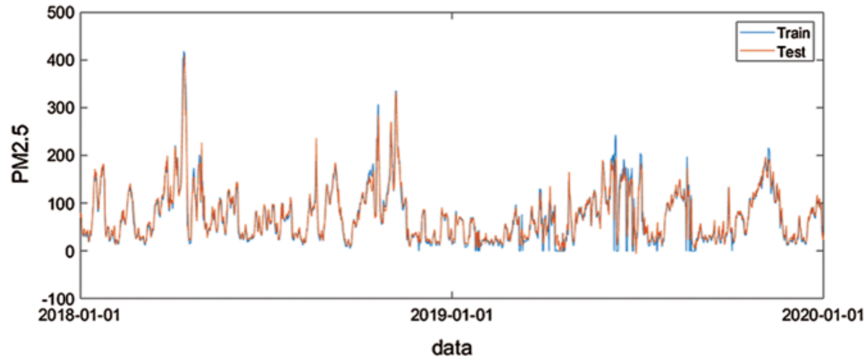
	Beijing	Shanghai	Nanjing	Guangdong	Hangzhou	Shenzhen
DTR	15.42	14.57	14.72	13.86	14.60	14.31
RF	13.05	13.62	13.56	12.22	12.99	12.47
SVR	14.57	14.23	14.51	14.85	14.85	13.89
RNN	8.84	8.12	9.49	8.93	8.24	9.63
LSTM	7.20	7.56	7.94	7.87	8.35	7.40
GRU	6.83	6.99	7.18	7.06	7.11	7.24
RNN-PTB	8.23	8.14	8.01	8.04	8.23	8.57
LSTM-PTB	7.12	7.23	7.20	7.38	8.04	7.17
GRU-PTB	**6.76**	**6.76**	**6.92**	**6.80**	**7.08**	**7.05**

Table 2: The false alarm rate of different methods in six cities or regions

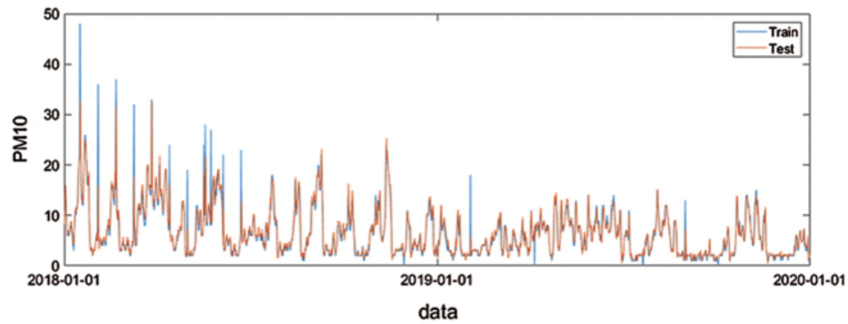
	Beijing	Shanghai	Nanjing	Guangdong	Hangzhou	Shenzhen
DTR	0.24	0.26	0.26	0.20	0.26	0.24
RF	0.22	0.22	0.24	0.21	0.22	0.22
SVR	0.22	0.23	0.23	0.20	0.21	0.22
RNN	0.16	0.18	0.16	0.17	0.17	0.17
LSTM	0.14	0.15	0.14	0.14	0.15	0.15
GRU	0.15	0.15	0.14	0.14	0.13	0.13
RNN-PTB	0.12	0.12	0.12	0.11	0.12	0.13
LSTM-PTB	**0.10**	**0.09**	0.10	**0.10**	0.10	0.10
GRU-PTB	**0.10**	0.10	**0.09**	**0.10**	**0.09**	**0.09**

Table 3: The $R2$ square of different methods in six cities or regions

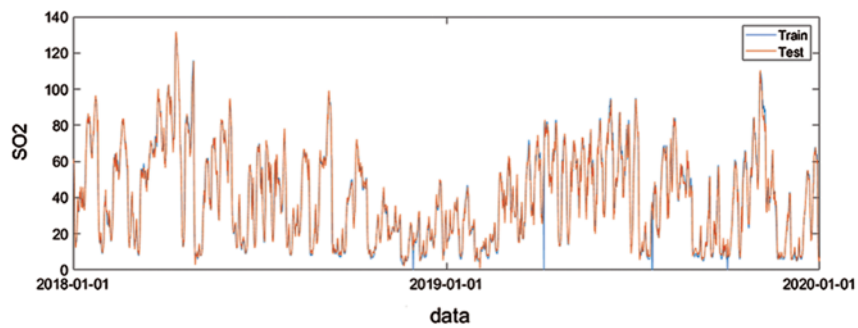
	Beijing	Shanghai	Nanjing	Guangdong	Hangzhou	Shenzhen
RNN	0.79	0.79	0.80	0.81	0.80	0.80
LSTM	0.82	0.84	0.82	0.86	0.83	0.82
GRU	0.82	0.82	0.82	0.81	0.82	0.82
RNN-PTB	0.83	0.83	0.84	0.84	0.84	0.83
LSTM-PTB	**0.85**	**0.86**	0.85	0.84	**0.85**	**0.86**
GRU-PTB	**0.85**	0.85	**0.86**	**0.86**	0.84	**0.86**


500
400
300
200
100
0
-100
PM2.5
Train
Test
2018-01-01
2019-01-01
2020-01-01
data

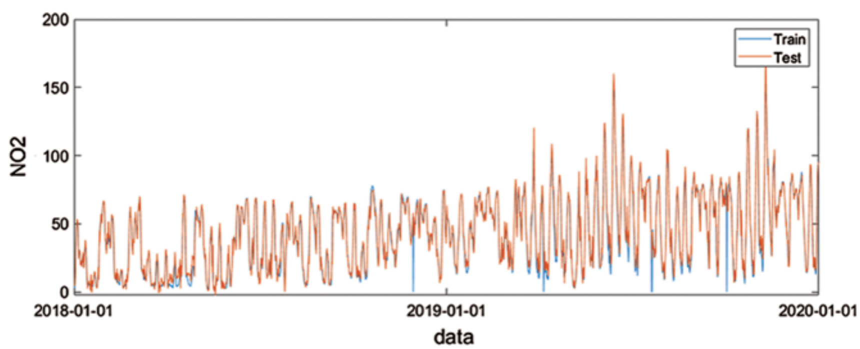
(a)


50
40
30
20
10
0
PM10
Train
Test
2018-01-01
2019-01-01
2020-01-01
data

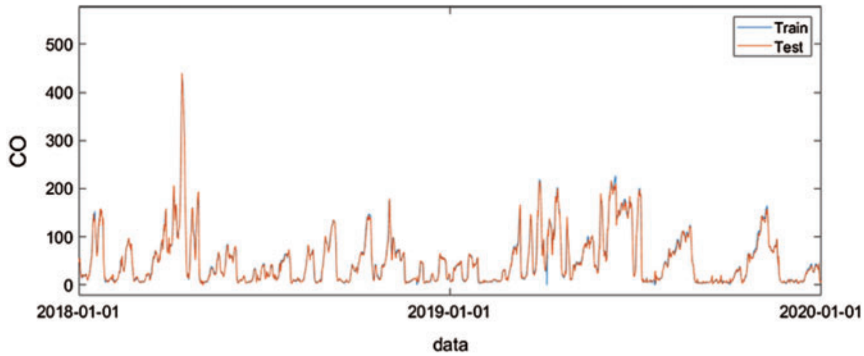
(b)


140
120
100
80
60
40
20
0
SO2
Train
Test
2018-01-01
2019-01-01
2020-01-01
data

(c)


200
150
100
50
0
NO2
Train
Test
2018-01-01
2019-01-01
2020-01-01
data

(d)



(e)

Figure 6: The forecasting of all features. (a) The forecasting of $PM_{2.5}$ value. (b) The forecasting of PM_{10} value. (c) The forecasting of SO_2 value. (d) The forecasting of NO_2 value. (e) The forecasting of CO value

5.3 Forecasting Results and Analysis

5.3.1 The Overall Results

To verify the accuracy of the PTB loss function, we compared the results on the original models and the variants combining the loss function proposed, and proved the superiority of the loss function by comparing the two results. First of all, we test on single feature prediction $PM_{2.5}$. In this experiment, $PM_{2.5}$ is seen as the regression value, the other data is seen as the training data. The results are shown in Tab. 1. From Tabs. 1 and 2 we can observe that: Firstly, compared with deep learning models, the traditional regression models cannot perform well in forecasting air quality data, which may be because these models ignore the sequential effects which are the most significant factor in the original data. Secondly, in deep learning methods, RNN achieved the worst result due to its hysteresis. However, when the PTB loss function replaced the original loss function, the performance got better, which to some extend shows the effectiveness of the loss function proposed in this paper. Due to this reason, LSTM and GRU were used in sequential data instead of RNN. Moreover, each RNN model and its derivative with PTB loss function performs better than those with original loss function. Thirdly, FAR was used to measure the stability of the model. Tab. 2 confirmed the stability of RNN models with PTB loss function, which may be because the trend loss controls that the predicted value of the model does not differ from the real value too much, which also shows the effectiveness of our loss function. Then, we start to verify the accuracy of multi-dimension prediction. In this experiment, all the air quality data of the day will be used as a unified label, and the training data remains the same. Considering that multiple labels cannot be measured by a single RMSE index, we use R-square index to measure the stability of the model. Moreover, the traditional method will not participate in this experiment because these methods cannot regress multiple labels well. The results of different deep neural networks are shown in Tab. 3.

From Tab. 3 we can observe that with the addition of PTB loss function, the fitting ability of the model is greatly improved. This is because the new loss function not only considers the accuracy of prediction value, but also considers the accuracy of prediction trend, similar to the first-order information and second-order information at the same time. It has been proved that this is beneficial to the accuracy and stability of the model prediction. Moreover, there is little difference in R_2 Square between each city in the same method, which means the method in this paper has good prediction effects in all kinds of weather conditions.

5.3.2 Prediction of Various Characteristics

In this section, we will show the specific prediction of each feature of this model, as shown in Fig. 6. From Fig. 6 we can find that our model gets accurate results in most forecasts, especially in $PM_{2.5}$ value, SO_2 value and CO value. However, the model is difficult to predict some values with large changes, such as in PM_{10} value. This shows that the prediction of data with large fluctuation in this paper is still not accurate. The possible reason is that the trend of data with large fluctuations is also large. However, the loss function proposed in this paper hopes that the trend of change will be gentle and stable. This is also our main work in the future.

6 Conclusion

This paper analyzed the air quality model of IoT monitoring which can realize prediction and analysis. Through the IoT monitoring and data prediction, a dynamic monitoring technology architecture integrating information collection, transmission, processing and prediction is constructed. This paper also proposed a novel loss function called predicted-trend-based (PTB) loss function which is used to replace the traditional loss function in RNN models. Through the double restriction of numerical accuracy and trend accuracy, the loss function proposed realized more accurate prediction. At the same time, this function was extended to multi-dimension feature prediction, therefore realized the stable prediction of the model. Then RNN related models were used on air quality data in Beijing from 2018-01-01 to 2020-01-01 and achieved good performance. The experiment also proved the stability of the model in multi-dimension prediction. Finally, we will focus on how to quantify and visualize the changing trend in future work.

Funding Statement: This work is supported by the National Natural Science Foundation of China under Grant 61972207, U1836208, U1836110, 61672290; the Major Program of the National Social Science Fund of China under Grant No. 17ZDA092, by the National Key R&D Program of China under Grant 2018YFB1003205; by the Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET) fund, China; by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) fund.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Schaffers, H., Komninos, N., Pallot, M., Trousse, B., Nilsson, M. et al. (2011). Smart cities and the future internet: towards cooperation frameworks for open innovation. *Future Internet Assembly,* 431–446.
2. Jia, M., Komeily, A., Wang, Y., Srinivasan, R. S. (2019). Adopting internet of things for the development of smart buildings: A review of enabling technologies and applications. *Automation in Construction, 101,* 111–126. DOI 10.1016/j.autcon.2019.01.023.
3. Shanmugapriya, P., Baskaran, J., Nayanatara, C., Kothari, D. (2019). IoT based approach in a power system network for optimizing distributed generation parameters. *Computer Modeling in Engineering & Sciences, 119(3),* 541–558. DOI 10.32604/cmes.2019.04074.
4. Kolehmainen, M., Martikainen, H., Ruuskanen, J. (2001). Neural networks and periodic components used in air quality forecasting. *Atmospheric Environment, 35(5),* 815–825. DOI 10.1016/S1352-2310(00)00385-X.

5. Zhang, Y., Bocquet, M., Mallet, V., Seigneur, C., Baklanov, A. (2012). Real-time air quality forecasting, part i: History, techniques, and current status. *Atmospheric Environment, 60,* 632–655. DOI 10.1016/j.atmosenv.2012.06.031.
6. Grell, G., Baklanov, A. (2011). Integrated modeling for forecasting weather and air quality: a call for fully coupled approaches. *Atmospheric Environment, 45(38),* 6845–6851. DOI 10.1016/j.atmosenv.2011.01.017.
7. Donnelly, A., Misstear, B., Broderick, B. (2015). Real time air quality forecasting using integrated parametric and non-parametric regression techniques. *Atmospheric Environment, 103,* 53–65. DOI 10.1016/j.atmosenv.2014.12.011.
8. Yahya, K., Zhang, Y., Vukovich, J. M. (2014). Real-time air quality forecasting over the Southeastern United States using WRF/Chem-MADRID: Multiple-year assessment and sensitivity studies. *Atmospheric Environment, 92,* 318–338. DOI 10.1016/j.atmosenv.2014.04.024.
9. Zhang, Y., Hong, C., Yahya, K., Li, Q., Zhang, Q. et al. (2016). Comprehensive evaluation of multiyear real-time air quality forecasting using an online-coupled meteorology-chemistry model over Southeastern United States. *Atmospheric Environment, 138,* 162–182. DOI 10.1016/j.atmosenv.2016.05.006.
10. Peng, H., Lima, A. R., Teakles, A., Jin, J., Cannon, A. J. et al. (2017). Evaluating hourly air quality forecasting in canada with nonlinear updatable machine learning methods. *Air Quality, Atmosphere & Health, 10(2),* 195–211. DOI 10.1007/s11869-016-0414-3.
11. Tamas, W., Notton, G., Paoli, C., Nivet, M. L., Voyant, C. (2016). Hybridization of air quality forecasting models using machine learning and clustering: An original approach to detect pollutant peaks. *Aerosol and Air Quality Research, 16(2),* 405–416. DOI 10.4209/aaqr.2015.03.0193.
12. Liu, B. C., Binaykia, A., Chang, P. C., Tiwari, M. K., Tsao, C. C. (2017). Urban air quality forecasting based on multi-dimensional collaborative support vector regression (SVR): a case study of Beijing-Tianjin-Shijiazhuang. *PLoS One, 12(7),* e0179763. DOI 10.1371/journal.pone.0179763.
13. Kalapanidas, E., Avouris, N. (2003). Feature selection for air quality forecasting: a genetic algorithm approach. *AI Communications, 16(4),* 235–251.
14. Jamal, A., Nodehi, R. N. (2017). Predicting air quality index based on meteorological data: A comparison of regression analysis, artificial neural networks and decision tree. *Journal of Air Pollution and Health, 2(1),* 27–38.
15. Lu, W. Z., Wang, W. J. (2005). Potential assessment of the "support vector machine" method in forecasting ambient air pollutant trends. *Chemosphere, 59(5),* 693–701. DOI 10.1016/j.chemosphere.2004.10.032.
16. Sagayaraj, S., Vetrivelan, N. (2018). Improving air quality management using gradient boosting based hierarchical temporal memory neural networks and fuzzy based classification based regression tree. *International Journal of Engineering & Technology, 2(9),* 12–17.
17. Miskell, G., Pattinson, W., Weissert, L., Williams, D. (2019). Forecasting short-term peak concentrations from a network of air quality instruments measuring PM2.5 using boosted gradient machine models. *Journal of Environmental Management, 242,* 56–64. DOI 10.1016/j.jenvman.2019.04.010.
18. Wang, D., Wei, S., Luo, H., Yue, C., Grunder, O. (2017). A novel hybrid model for air quality index forecasting based on two-phase decomposition technique and modified extreme learning machine. *Science of the Total Environment, 580,* 719–733. DOI 10.1016/j.scitotenv.2016.12.018.
19. Sahin, Ü. A., Bayat, C., Uçan, O. N. (2011). Application of cellular neural network (CNN) to the prediction of missing air pollutant data. *Atmospheric Research, 101(1–2),* 314–326. DOI 10.1016/j.atmosres.2011.03.005.
20. Du, H., Du, Y., Huang, S., Tang, Y. J., Xie, B. G. et al. (2019). Air quality forecasting based on dynamic blending. U.S. Patent 10,330,655.
21. Lin, Y., Zhao, L., Li, H., Sun, Y. (2018). Air quality forecasting based on cloud model granulation. *EURASIP Journal on Wireless Communications and Networking, 2018(1),* 106. DOI 10.1186/s13638-018-1116-3.

22. Lawrence, S., Giles, C. L., Tsoi, A. C., Back, A. D. (1997). Face recognition: A convolutional neural-network approach. *IEEE Transactions on Neural Networks, 8(1),* 98–113. DOI 10.1109/72.554195.
23. Yi, X., Zhang, J., Wang, Z., Li, T., Zheng, Y. (2018). Deep distributed fusion network for air quality prediction. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 965–973. New York, USA: ACM.
24. Rajalakshmi, M., Rengaraj, R., Bharadwaj, M., Kumar, A., Naren Raju, N. et al. (2018). An ensemble based hand vein pattern authentication system. *Computer Modeling in Engineering & Sciences, 114(2),* 209–220.
25. Mikolov, T., Karafiát, M., Burget, L., Cernock, J., Khudanpur, S. (2010). Recurrent neural network based language model. *Eleventh Annual Conference of the International Speech Communication Association*, pp. 1045–1048. Makuhari, Chiba, Japan: Interspeech.
26. Tsai, Y. T., Zeng, Y. R., Chang, Y. S. (2018). Air pollution forecasting using RNN with LSTM. *IEEE International Conference on Dependable, Autonomic & Secure Computing, International Conference on Pervasive Intelligence & Computing, International Conference on Big Data Intelligence & Computing & Cyber Science & Technology Congress*, pp. 1074–1079. Athens, Greece: IEEE.
27. Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9(8),* 1735–1780. DOI 10.1162/neco.1997.9.8.1735.
28. Fan, J., Li, Q., Hou, J., Feng, X., Karimian, H. et al. (2017). A spatiotemporal prediction framework for air pollution based on deep RNN. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 4,* 15–22. DOI 10.5194/isprs-annals-IV-4-W2-15-2017.
29. Kim, M. H., Kim, Y. S., Lim, J., Kim, J. T., Sung, S. W. et al. (2010). Data-driven prediction model of indoor air quality in an underground space. *Korean Journal of Chemical Engineering, 27(6),* 1675–1680. DOI 10.1007/s11814-010-0313-5.
30. Afanador, N., Tran, T., Buydens, L. (2013). Use of the bootstrap and permutation methods for a more robust variable importance in the projection metric for partial least squares regression. *Analytica Chimica Acta, 768,* 49–56. DOI 10.1016/j.aca.2013.01.004.
31. Athira, V., Geetha, P., Vinayakumar, R., Soman, K. (2018). Deepairnet: applying recurrent networks for air quality prediction. *Procedia Computer Science, 132,* 1394–1403. DOI 10.1016/j.procs.2018.05.068.
32. Chung, J., Gulcehre, C., Cho, K., Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv: 1412.3555.
33. Wang, B., Kong, W., Li, W., Xiong, N. N. (2019). A dual-chaining watermark scheme for data integrity protection in internet of things. *Computers Materials & Continua, 58(3),* 679–695. DOI 10.32604/cmc.2019.06106.
34. Minoli, D., Sohraby, K., Occhiogrosso, B. (2017). IoT considerations, requirements, and architectures for smart buildings energy optimization and next-generation building management systems. *IEEE Internet of Things Journal, 4(1),* 269–283. DOI 10.1109/JIOT.2017.2647881.
35. Chen, H., Jin, H., Wu, S. (2015). Minimizing inter-server communications by exploiting self-similarity in online social networks. *IEEE Transactions on Parallel and Distributed Systems, 27(4),* 1116–1130. DOI 10.1109/TPDS.2015.2427155.