IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 12, 2021, accepted June 26, 2021, date of publication June 29, 2021, date of current version July 6, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3093430

Air Quality Prediction Based on Integrated Dual LSTM Model

HONGQIAN CHEN[1], MENGXI GUAN[1], AND HUI LI[2]

[1]Beijing Key Laboratory of Big Data Technology for Food Safety, School of Computer Science and Engineering, Beijing Technology and Business University, Beijing 100048, China
[2]Management College, Beijing Union University, Beijing 100101, China

Corresponding author: Hui Li (lihui@buu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 31701517, in part by the Beijing Philosophy and Social Science Foundation under Grant 17GLC060 and Grant 20GLB032, in part by the Academic Research Project of Beijing Education Commission under Grant SZ202111417021, and in part by the Academic Research Projects of Beijing Union University under Grant ZB10202005 and Grant JS10202006.

This work did not involve human subjects or animals in its research.

ABSTRACT Air quality prediction is an important reference for meteorological forecast and air controlling, but over fitting often occurs in prediction algorithms based on a single model. Aiming at the complexity of air quality prediction, a prediction method based on integrated dual LSTM (Long Short-Term Memory) model was proposed in this paper. Firstly, the Seq2Seq (Sequence to Sequence) technology is used to establish a single-factor prediction model which can obtain the predicted value of each component in air quality data, independently. Each component of air quality is regarded as time series data in the forecasting process. Then, the LSTM model with attention mechanism is used as the multi-factor prediction model. The influencing factors of air quality, like the data of neighboring stations and weather data, are considered in the model. Finally, XGBoosting (eXtreme Gradient Boosting) tree is used to integrate two models. The final prediction results can be obtained by accumulating the predicted values of the optimal subtree nodes. Through evaluation and analysis using five evaluation methods, the proposed method has better performance in terms of error and model expression power. Compared with other various models, the precision of prediction data has been greatly improved in our model.

INDEX TERMS Air quality prediction, integrated dual model, LSTM model with attention mechanism, Seq2Seq technology, XGBoosting tree.

I. INTRODUCTION

With the improvement of the level of industrialization, the exhaust gas produced by a large number of factories and cars continues to increase, resulting in the air pollution rises seriously. Air quality has a great impact on people's daily life. Accurate prediction of air quality has become an important measure to control air pollution and improve the air quality.

Air quality data has been widely concerned in the world. Time series data prediction method, like traditional machine learning methods [1]–[6] and time series prediction models [7]–[9], is often used for air quality prediction. However, the existing air quality prediction methods cannot effectively capture the complex nonlinearity of air quality, like PM2.5 concentrations. The prediction models based on deep learning [10]–[15] can extract the features existing in the air quality data and can achieve higher prediction accuracy. Some methods [16]–[26] simulate the temporal and spatial dependence of air quality data at the same time. But widely-used machine learning methods often suffer from high variability in performance in different circumstances. Air quality is affected by many factors, such as temperature, wind power and spatial relationship. As a result, the common single model prediction method is difficult to obtain certain and accurate prediction results. Integrating multiple models to predict air quality [27]–[35] is a type of method that appears in the latest literature, and it is also the source of our ideas in this article. The integrated model can significantly improve the forecasting ability compared with existing models. However, how to integrate the advantages of multiple models according to the characteristics of the data set is still an important topic that needs to be studied.

The associate editor coordinating the review of this manuscript and approving it for publication was Dongxiao Yu.

The supporting evidences of the model proposed in this paper are mainly the characteristics of air quality data. Air quality data is a type of data set with complex and various characteristics. According to our in-depth analysis of air quality data, we found that air quality data has many types of characteristic, which are mainly reflected in two aspects.

Firstly, in some peaceful weather with sunshine, stable air pressure, and breeze, meteorological factors have a particularly small impact on the changes in air quality data. In this situation, the air quality data basically conforms to the characteristics of time series data, like stable change trend and periodicity, etc. Figure 1 shows the correlation between air quality data of the current moment and the data of 24 hours ago or 48 hours ago in peaceful weather. The air quality values at the current moment are in vertical axes of Fig. 1(a) and Fig. 1(b), while the values of 24 hours ago and 48 hours ago are in the horizontal axes, respectively. A preliminary conclusion can be drawn from Figure 1 that the air quality data shows a greater correlation with that of 24/48 hours ago in the calm weather.

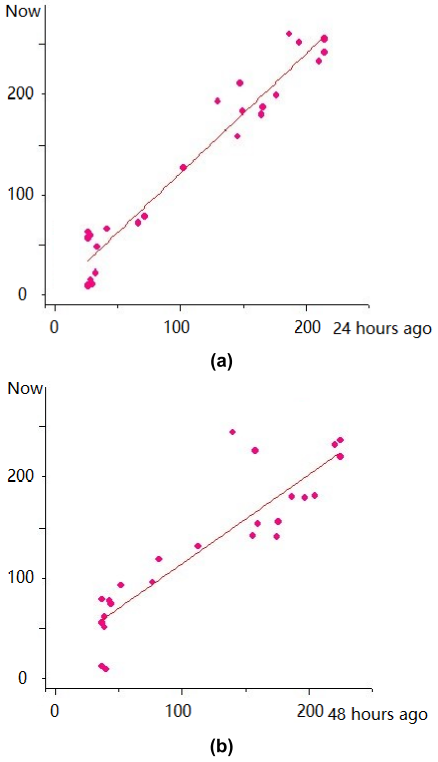

FIGURE 1. The correlation between the air quality data of the current moment and the data of 24 hours ago or 48 hours ago in peaceful weather, (a) the correlation between the data of the current moment and 24 hours ago, (b) the correlation between the data of the current moment and 48 hours ago.

Secondly, in other types of weather, like strong winds and heavy pollution in some areas, meteorological factors have a greater impact on the air quality data. In this case, the air quality data shows the characteristics of the spatial data. Figure 2 shows the clustering results of air quality data for

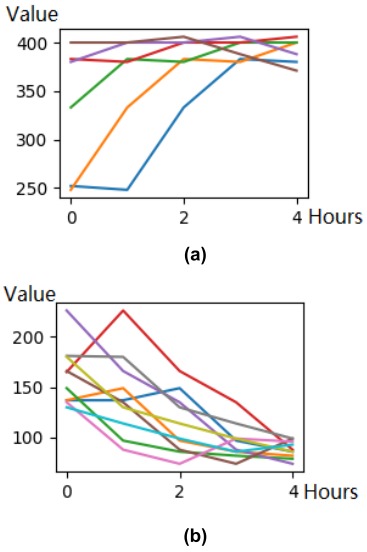

FIGURE 2. The clustering results of air quality data for different stations in windy weather, (a) the data of adjacent stations tend to be similar high values; (b) the data of adjacent stations tend to be similar low values.

different stations in windy weather. The results in Fig. 2 are obtained using the K-Means clustering method based on DTW distance, in which the sliding window size is set as 5 hours. From Fig. 2, it can be found that the air quality data of different stations under windy weather will gradually tend to be similar in values.

According to the in-depth analysis of the different characteristics of the air quality data, the research work of this article established two independent models from both time dimension and spatial dimension. XGBoosting technology was used to implement the integration of dual model. First, the single-factor model for each component was established in the time dimension. Single factors, like PM2.5, are used as the input of single-factor models. The forecast results are obtained by using the characteristics in the time dimension. Then, the multi-factor model is established in the spatial dimension. Multiple factors, such as data from the current station and surrounding stations, as well as weather data, are selected as the input of the multi-factor model together. The prediction results are obtained according to the spatial characters. Finally, XGBoosting tree takes the output of the single-factor time model and the multi-factor spatial model as input. The optimized predicted value is obtained by calculating the weight of each leaf node and accumulating its predicted value. The proposed prediction method can adapt to the complex changing characteristics of the data set.

In summary, the contributions of this article are shown in the following aspects.

(1) Two types of forecasting models were adopted separately from the time dimension and spatial dimension to better meet the different types of characteristics of air quality data. Single-factor forecasting models were used to obtain the characteristics of each component of the data from the time dimension. Multi-factor forecasting models were used

to acquire the characteristics of influence among data of the current weather station and surrounding weather stations, as well as weather data from the space-time dimension.

(2) Meteorological data, as the most influential factor, were combined into the training process of XGBoosting tree to reach the optimal subtree. The confidence weights of the prediction results between the two models were set based on the optimal subtree, which can effectively improve the prediction ability.

The method in this article not only can be used for the prediction of air quality data, but also can be adopted to predict and analyze various data sets with multiple data characteristics. For this type of data set, a variety of model stacking methods can be used to improve the accuracy of the prediction results. In the process, the influencing factors of feature changes, like meteorological data, can be used as a reference for the model stacking process.

The rest of this paper is organized as follows. The related work will be discussed in Part II. The materials and the detailed description of our proposed method will be described in Part III. Part IV will display the experimental results of the integrated model, as well as the comparisons and analysis between our model and other methods. Finally, we will present the conclusion of our work and the direction of further researches.

II. RELATED WORK

Many researchers have made great contributions to the problem of air quality prediction in recent years. Various patterns and basic trends in air quality are identified by quantitative researches combining the latest technology. The main technologies and implementation methods in these achievements include the following categories.

A. PREDICTION BASED ON CLASSIC MACHINE LEARNING METHODS

Classic machine learning techniques, such as regression analysis, principal component analysis, BP (Back Propagation) network and artificial neural network, were once the mainstream method of air quality prediction. Petr and Vladimir [1] designed a model based on feed-forward neural networks of perceptron and fuzzy inference systems for air quality prediction. Kang and Qu [2] established BP neural network based on genetic simulated annealing algorithm optimization to predict air quality. Wang *et al.* [3] trained a BP neural network based on the historical monitoring data of air pollutants to predict PM2.5 mass concentration. Rajput and Sharma [4] represented the variation of AQI (Air Quality Index) with a multivariate regression model. Major parameters, such as ambient temperature, relative humidity and bar pressure, were considered in the regression model for AQI computation. Mahajan *et al.* [5] clustered the monitoring stations based on the geographical distance to reduce the forecasting errors and achieve acceptable forecast results of PM2.5 concentrations. Li *et al.* [6] built a dynamic evaluation model for forecasting the air quality data based on the fuzzy mathematical synthetic evaluation. The future air quality status would be built by the fuzzy synthetic assessment model based on entropy weighing method and whose results showed that the proposed evaluation model is a practical tool.

These classic methods and models have some advantages, such as simple algorithms, easy-to-understand processing and acceptable prediction results. These methods can well predict the trend of air quality changes. However, it is difficult to obtain specific accurate forecast values of air quality.

B. PREDICTION BASED ON TIME SERIES MODEL

The second type of methods are the prediction methods based on time series data. These methods regard continuous air quality data over a period of time as time series data, and obtain specific forecast values of air quality. Liu *et al.* [7] proposed an attention-based air quality predictor (AAQP), which used an n-step recurrent prediction to solve the problem of error accumulation produced in recurrent processing. Gu *et al.* [8] proposed a heuristic recurrent air quality predictor (RAQP) to exploit the meteorological factors and air pollutant concentration data which have strong influences on air quality of the next adjacent moment. Benhaddi and Ouarzazi [9] built a WaveNet architecture to forecast the conditional multivariate time series data. The architecture is composed of stacked residual convolutions which used parameterized skip connections to catch early trends in a large scope in the time series history.

In time series prediction methods, like RNN (Recurrent Neural Network), future data can be predicted according to the rules of data changes. However, when facing data with too long sequence information, gradient disappearance or gradient explosion may be occurred, which will lead to inaccurate prediction results.

C. PREDICTION BASED ON LSTM MODEL

The third type of methods are the LSTM-based prediction models. LSTM is an improved algorithm of RNN network, which can memorize long-term information in sequence data. Song *et al.* [10] proposed LSTM-Kalman time prediction model. The model stores the information contained in the pre-order data by using LSTM, while adjusts the basic time data sequence by Kalman filtering. Wang *et al.* [11] established CT-LSTM by combining CT (Chi-square Test) and LSTM. CT is used to determine the influencing factors of air quality which can help improving the accuracy and performance of prediction. Jianhui *et al.* [12] proposed the LSTM-FWA model based on LSTM and FWA (FireWorks Algorithm). The model is optimized with temporal, spatial, spatio-temporal techniques respectively. Qin *et al.* [13] integrated a hyperbolic model to predict PM2.5 concentrations as time series based on CNN (Convolutional Neural Network) and LSTM network. CNN network is used to extract features of input data, while LSTM network is used to consider the time dependence of air pollutants. Li *et al.* [14] introduced the attention mechanism into the LSTM to capture the importance degrees of featured states at different times. The model

can predict the PM2.5 concentrations over the next 24 hours by using air quality data. Luo *et al.* [15] established the BiLSTM (bidirectional long short-term memory) network, in which an EMD (empirical mode decomposition) step is introduced to reduce error accumulation in PM2.5 multi-step prediction.

LSTM-based model solves the problems of gradient explosion and gradient disappearance in RNN, and has a faster learning speed. Therefore, the LSTM-based model can effectively obtain better prediction results. However, it is still difficult to obtain a high accuracy rate for data prediction because of too many factors affecting air quality changes.

D. PREDICTION ON SPATIO-TEMPORAL FACTORS

The fourth type of methods are the prediction methods which treat air quality data as spatio-temporal data in analyzing. A variety of influencing factors are considered from the time dimension and space dimension to improve the predictive ability. Belavadi *et al.* [16] used a scalable architecture to monitor and gather real-time air pollutant concentration data from wireless sensor network in various places and to forecast future air pollutants concentrations. Sun *et al.* [17] established a spatio-temporal GRU-based (Gated Recurrent Units) prediction framework which takes the spatial information into consideration to predict PM2.5 concentrations in the hour scale. Xiangyu *et al.* [18] established STA-LSTM neural network based on LSTM, in which a STA (Spatio-Temporal Attention) mechanism was introduced to capture the relative influence of surrounding stations on the prediction area. Qin *et al.* [19] predicted the short-term air quality based on KNN (K-nearest neighbor) and LSTM. The training processes are constructed on the AQI sequences of the space-related monitoring stations selected by KNN algorithm.

Zhao *et al.* [20] used a fully connected neural network to combine the spatial information of surrounding stations, and achieve an accurate prediction of urban PM2.5 contaminations over 48 hours. Ping-Wei *et al.* [21] implemented the air quality forecasting for up to 48 hours using a combination of multiple neural networks. Altitude information and meteorology data are combined with the air quality data from the previous few hours to improve the forecasting ability of the model. Qi *et al.* [22] integrated Graph Convolutional Networks and LSTM networks (GC-LSTM) to model and forecast the spatio-temporal variation of PM2.5 concentrations. The historical observations on different stations are constructed as spatio-temporal graph series for 72-hour predictions.

Seng *et al.* [23] proposed a multi-output and multi-index supervised learning (MMSL) model to integrate the concentration data, the meteorological data, and the gaseous pollutant data of the present monitoring station and its nearest neighbor stations of the same period. LSTM was used for training to obtain the predicted values of air quality pollution indicators. Zhou *et al.* [24] proposed a Deep Multi-output LSTM (DM-LSTM) neural network model which integrates three deep learning algorithms for the air quality forecasting. The model extracts the key factors of complex spatio-temporal relations to reduce error accumulation and propagation in the multi-step air quality forecasting. Yan *et al.* [25] established a multi-time, multi-site forecasting model based on spatiotemporal clustering for air quality forecasting. The spatiotemporal distribution characteristics were introduced into the forecasting processing. The CNN-LSTM and the LSTM model were proved more suitable for the multiple-hour forecasting in the comparing experiments. Xu and Yoneda [26] proposed the LSTM auto-encoder multitask learning model to predict PM2.5 time series in multiple locations. The model utilized the multi-layer LSTM networks to simulate the spatiotemporal characteristics of urban air pollution particles. The pattern of urban meteorological systems and the dynamical relationship among multiple key pollution time series were adopted to provide important auxiliary information for PM2.5 time-series prediction in the model.

In the prediction method combining time and space factors, the data continuity between geographic locations can be used to further improve the accuracy of prediction. But widely-used machine learning methods often suffer from high variability in performance in different circumstances. The best machine learning method varies between regions and times, making method selections difficult.

E. PREDICTION ON INTEGRATED MODEL

Leizhi *et al.* [27] proposed a model stacking approach where the outputs of five widely-used individual machine learning models are taken as input features of the ensemble model. The models which are selected include multiple linear regression, partial least square, sparse partial least square, random forest, and Bayesian network. The model stacking approach was able to generate more reliable prediction results than others models. Guoyan *et al.* [28] implemented an integration method of GRU neural network based on empirical mode decomposition (EMD-GRU). The sub-sequences extracted from the time series of multiple stations are added to obtain the prediction results of PM2.5 concentrations. Yue-Shan *et al.* [29] proposed a stacking-based ensemble learning scheme to integrate various forecasting models together. Pearson correlation coefficient is adopted to calculate the correlation between different models in the stacking-based scheme. Jiaqi *et al.* [30] designed an attention-based parallel network (APNet), which uses a Bi-LSTM parallel module to extract the periodic characteristics of PM2.5 concentrations from both previous and posterior directions.

The hybrid model based on model stacking can significantly improve the forecasting accuracy compared with existing models. Bai *et al.* [31] proposed an ensemble long short-term memory neural network (E-LSTM) model. The ensemble empirical mode decomposition were employed in the feature extraction and multi-modal feature estimated integration. The multiple LSTMs structure in E-LSTM model achieved better forecasting performance than the single

LSTM structure. Wei [32] established an accurate wind speed prediction model for future typhoons based on stacked long short-term memory (SLSTM). The experimental results showed that the SLSTM yielded more accurate results than MLP and DRNN. Moniz and Krueger [33] proposed Nested LSTMs (NLSTM), which add depth to LSTMs via nesting as opposed to stacking. NLSTM outperformed both stacked and single-layer LSTMs with similar numbers of parameters. Jin *et al.* [34] proposed a novel deep learning framework combining multiple nested long short-term memory networks (MTMC-NLSTM) for accurate AQI forecasting. The federated learning in the framework strengthened the performance of obtaining more accurate prediction results.

Tree boosting is a highly effective and widely used machine learning method. XGBoost [35] implements machine learning algorithms under the Gradient Boosting framework. XGBoost improves the accuracy of prediction results by adding trees one by one. In XGBoosting tree, each addition of subtrees will improve accuracy of the prediction results, and finally the most accurate prediction results are obtained.

III. INTEGRATED DUAL LSTM MODEL METHOD

Aiming at improving the accuracy of air quality prediction, a multi-model integration method was proposed in the paper. The detailed process of air quality prediction based on integrated dual LSTM model in this paper is shown in Fig. 3.

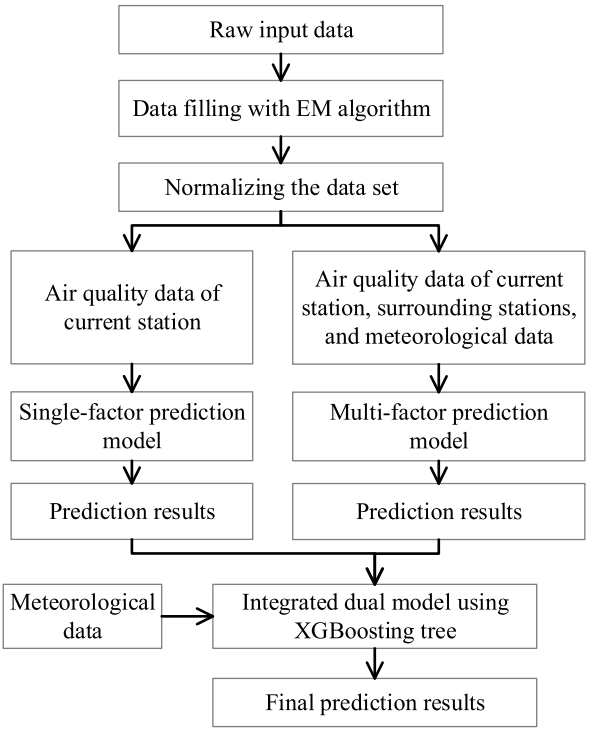

FIGURE 3. The process flow of the air quality prediction based on LSTM model stacking method.

The whole framework of the method is composed of four main parts. The first part is preprocessing the data, including filling in missing values and normalizing all the data. The second part is constructing a single-factor prediction model based on Seq2Seq to forecast each influencing factor individually. The encoding and decoding structure consists of LSTM units. The third part is constructing a multi-factor prediction model based on the attention mechanism and Seq2Seq. Various influencing factors, such as historical data of the current station and surrounding stations, as well as weather data, are used as input of the multi-factor model. The encoding and decoding structure are composed of two layers of LSTM units. Finally, XGBoosting tree is used to integrate the prediction results of the single-factor model and the multi-factor model. The best prediction results obtained through regression calculation are used as the final prediction value of the air quality data.

A. DATA SET AND DATA PREPROCESSING

1) DATA SET DESCRIPTION

We collected hourly data from multiple air quality monitoring stations in Beijing from 2013 to 2018. The air pollutants in the data include PM2.5, PM10, NO_2, CO, O^3 and SO_2. In addition, the data set contain meteorological data of the same period, including weather type, temperature, pressure, humidity, wind speed and wind direction. The unit of air pollutants is g/m^3. Weather types mainly include sunny, snowy, cloudy, light rain, heavy rain, and blowing sand. The unit of temperature is Celsius (°C). The unit of air pressure is hectopascals (hPa). Humidity refers to the content of water vapor in the air, expressed as percentage (%). The unit of wind speed is meters per second (m/s). The wind direction is defined by the clockwise angle from the north. For example, the direction of the wind blowing from the south is 180 degrees, and the direction of the wind blowing from the east is 90 degrees. This data set mainly come from the Meteorological Data Center of China Meteorological Administration.

2) DATA FILLING WITH EM ALGORITHM

When the monitoring station acquires air quality data, there may be missing values in the air quality data due to sensor device failures or network problems. Dataset containing null values may lead to unreliable output. Therefore, the data set needs to be interpolated and filled with data before prediction.

Traditional data filling methods mainly include deletion, mean filling and neighbor replacement, etc. EM algorithm [36] was selected to complete the null value in this article. EM algorithm is a data filling algorithm proposed by Dempster, Arthur P. of Harvard University in 1977. It is a classic data filling algorithm for incomplete data sets. The most special feature of EM algorithm is to find the maximum likelihood estimation or maximum posteriori estimation of parameters in the probability model, where the probability model depends on unobservable hidden variables.

The main reason we chose EM algorithm is that the data set completed by the EM algorithm can maintain a similar distribution probability with the original data set. It can make

the filled data keep the distribution probability close to the original data. The theoretical basis for applying the EM algorithm to the filling of air quality data sets is as follows. Air quality data have many different characteristics, as described in the part of Introduction. But the distribution probability of data with different pollutant concentrations is basically stable. For example, the proportion of PM2.5 data between 100 and 120 in all data is basically stable for a period of time. According to the conclusion of our previous data analysis, the air quality data set in this article conforms to the Gaussian normal distribution, and the Gaussian function is used as the distribution function in the EM algorithm.

Specifically, EM algorithm includes E steps and M steps, where the E step seeks the maximum likelihood estimations of the sample, and the M step seeks the maximum likelihood results.

For given mutually independent data samples $\{x^1, \ldots, x^m\}$, the goal of EM algorithm is to find the implicit category z of each sample, so that the maximum likelihood function $p(x, z)$ is the largest. The goal of E step is to calculate the log likelihood expectation function based on the sample x, and estimate the maximum likelihood z of the hidden variables, which can be expressed as Equation (1),

$$Q_i\left(z^{(i)}\right) = p\left(z^{(i)}|x^{(i)};\theta\right) \tag{1}$$

where, x is the given sample $\{x^1, \ldots, x^m\}$, and z is the hidden variable. $p\left(z^{(i)}|x^{(i)};\theta\right)$ is the posterior probability of given sample $x^{(i)}$ and the parameter θ. $Q_i\left(z^{(i)}\right)$ can be selected by the posterior probability of $z^{(i)}$.

For the M step, the expected result of the E step likelihood function will be maximized. The maximum lower bound of the log likelihood function is selected in the E step, as shown in Equation (2),

$$\theta = \underset{\theta}{\operatorname{argmax}} \sum_i \sum_z^{(i)} Q_i\left(z^{(i)}\right) \log \frac{p\left(z^{(i)}|x^{(i)};\theta\right)}{Q_i\left(z^{(i)}\right)} \tag{2}$$

The new expected value will be obtained by executing repeatedly E step and M step until convergence. The new expected value will be used to complete missing values.

3) NORMALIZING THE DATA SET

The value ranges of different components of air quality data vary greatly because of differences of their measurement units. The data should be normalized to eliminate the influences of different numerical ranges. Normalization of data set can also speed training up and improve prediction accuracy.

The normalization processing of the original sample data is shown in Equation (3). Where, X is the original data, X_{min} represents the minimum value in the original data, and X_{max} represents the maximum value in the original data.

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{3}$$

Each component in the processed data set will be distributed in the interval [0, 1]. The normalized data set is divided into training data set and test data set at a ratio of 8:2.

B. SINGLE-FACTOR PREDICTION MODEL BASED ON LSTM

The structure of a single-factor prediction model includes an input layer, hidden layers and an output layer. Each historical concentration data of the six pollutants, PM2.5, PM10, NO_2, CO, O^3, SO_2, can be used as the inputs of the model. The structure of the hidden layer is a Seq2Seq module including encoding and decoding parts. The output of the model is the corresponding prediction values of the pollutants in the input data. The structure of single-factor prediction model can be described as Fig. 4.

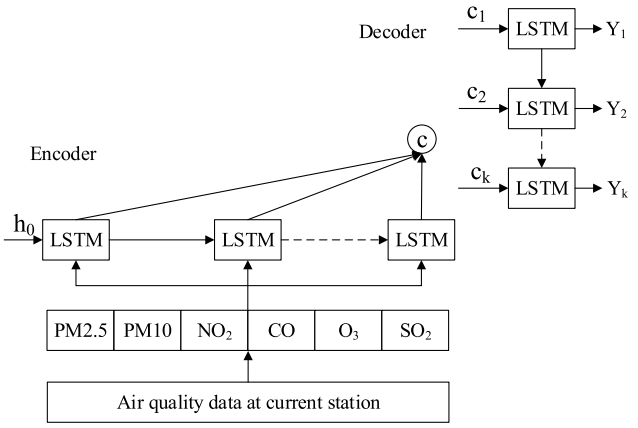

FIGURE 4. Structure of single-factor prediction model.

In terms of time dimension, air quality data sequence can be divided into trend, periodicities and residual items. Our selected periodicities include daily, weekly, monthly, quarterly and yearly period. Time features are related to cycles, for example, 23 o'clock and 0 o'clock should be adjacent in terms of value. Cos or Sin function can be used to encode them, as shown in Equation (4).

$$vhour = \cos\left(2 * pi * \frac{hour}{24}\right) \tag{4}$$

Taking PM2.5 data prediction as an example, the dimension of the input layer is $t*1$, where t represents the number of selected historical data in the input of the model. The number of input data in the model can be shown in Equation (5).

$$n_{input} = 24 + 6 * x \tag{5}$$

Among them, 24 means that the input data of the single-factor model contains historical data of 24 hours before the current moment; x means that it also contains x hours one day ago, x hours two days ago, x hours one week ago, x hours a month ago, x hours a quarter ago, and x hours a year ago.

The hidden layer adopts the Seq2Seq module and uses the encoder-decoder structure. Both of the encoder and decoder are a structure consisting of multi-layered LSTM units. The encoder of the single-factor model consists of two layers of RNN, and each layer is equipped with 64 LSTM neural units. For a given sequence $x = \{x_1, x_2, \ldots, x_t\}$, the specific formula for the encoder is shown as Equation (6),

$$h_t = f\left(h_{(t-1)}, x_t\right) \tag{6}$$

where, t is the current time, h_t is the hidden state at time t, and f is the LSTM encoder.

After encoding, the context vector c, obtained by combining all h_t, can be used to express important features of the input data.

The decoder consists of two layers of RNN, and each layer is equipped with 64 LSTM units. The specific formula of the decoding part is shown as Equation (7),

$$\begin{aligned} h_t &= f\left(h_{(t-1)}, y_{(t-1)}, c\right) \\ p\left(y_t|y_{(t-1)}, y_{(t-2)}, \ldots, y_1, c\right) &= g\left(h_t, y_{(t-1)}, c\right) \end{aligned} \tag{7}$$

where, h_t is the output of the encoder, c is the context vector produced in the encoder, $f()$ and $g()$ are non-linear activation functions of LSTM units.

The dimension of the output layer is $1 * k$, where k represents the model can produce the air quality prediction results from 1 hour to k hours in the future.

The training parameter settings of the model are shown in Table 1.

TABLE 1. The training parameter settings of the single-factor model.

Hyperparameter	Specific settings
Number of iterations	1000
Time step	48
Learning rate	0.01
Loss function	L2 loss
Batch size	24
Input dimension	48×6
Output dimension	6×1
Encoder layers	2
Encoder neural unit per layer	64
Decoder layer number	2
Decoder neural unit of each layer	64

C. MULTI-FACTOR PREDICTION MODEL BASED ON LSTM WITH ATTENTION MECHANISM

In terms of spatial dimension, there is a data correlation between each monitoring station and its surrounding stations. In addition, air quality data will be affected by weather factors. A multi-factor forecasting model is established based on LSTM with attention mechanism. The input data of the model include the air quality data of the current station and the neighboring stations, as well as meteorological data. The hidden layer structure consists of Seq2Seq modules with attention mechanism. The Seq2Seq module is an encoder-decoder structure. Two attention mechanisms are added before the encoder and decoder respectively. The output data are the predicted values of air quality data.

The structure of multi-factor prediction model can be described as Fig. 5.

For the given sequence $x^k = \left(x_1^k, x_2^k, \ldots, x_T^k\right)$, the encoder part constructs a feed forward neural network with an attention mechanism. The specific formula is

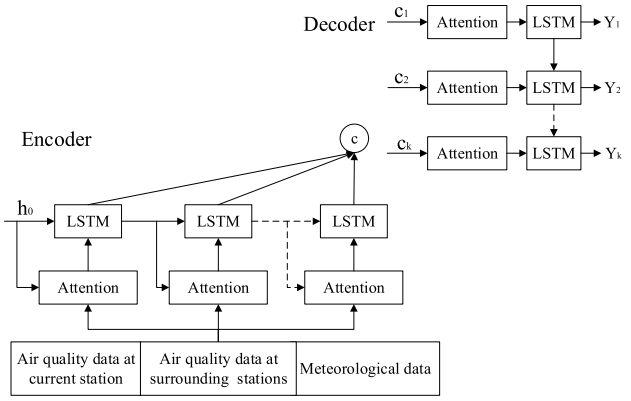

FIGURE 5. Architecture of multi-factor prediction model.

shown as Equation (8),

$$e_t^k = v_e \tanh\left(w_e\left[h_{(t-1)}; s_{(t-1)}\right] + u_e x^k\right) \tag{8}$$

where, $h_{(t-1)}$ and $s_{(t-1)}$ are the hidden state and neuron state of the previous section, and w_e and u_e are the learning parameters. Softmax function is used to ensure that the sum of the weights a_t^k is equal to 1. The calculation method of the weight a_t^k of e_t^k is shown as in Equation (9),

$$a_t^k = \frac{exp\left(e_t^k\right)}{\sum_{i=1}^{n} xp\left(e_t^i\right)} \tag{9}$$

where, a_t^k is the attention weight of the k sequence at time t. According to these attention weights, a new sequence with attention weights can be calculated, as shown in Equation (10).

$$\tilde{x}_t = \left(a_t^1 x_t^1, a_t^2 x_t^2, \ldots, a_t^n x_t^n\right) \tag{10}$$

The next multi-layer LSTM network is used for encoding, whose encoding method is shown as Equation (11).

$$h_t = f\left(h_{(t-1)}, \tilde{x}_t\right) \tag{11}$$

where, f is the LSTM unit, $\tilde{x}_t$ is the new sequence with attention weights, and h_t is the hidden state at time t.

For the state h_t obtained after encoding, an attention mechanism is added, as shown in Equation (12),

$$\lambda_t^i = v_d \tanh\left(w\left[d_{(t-1)}; s_{(t-1)}\right] + u_d h_i\right) \tag{12}$$

where, λ_t^i is the weight of the k sequence at time t, $d_{(t-1)}$ and $s_{(t-1)}$ are the hidden state and neuron state of the previous section, and w and u_d are the learning parameters. Softmax function is used to ensure that the sum of all the weights of λ_t^i is equal to 1. The calculation method for the weight β_t^i of λ_t^i is shown as in the Equation (13),

$$\beta_t^i = \frac{exp\left(\lambda_t^i\right)}{\sum_{j=1}^{T} xp\left(\lambda_t^j\right)} \tag{13}$$

where, β_t^i is the weight of the encoder output state at time i. The weighted sum of the attention weight and the hidden state

of the encoder is used as the context vector, and its calculation method is shown as in Equation (14),

$$c_t = \sum_{i=1}^{T} \beta_t^i h_i \tag{14}$$

where, c_t is the context vector, $\{h_1, h_2, \ldots h_T\}$ is the hidden state of the encoder, and β_t^i is the attention weight.

In the decoder, the context vector c_t is combined with a target sequence $(y_1, y_2, \ldots, y_{T-1})$, as shown in Equation (15),

$$\tilde{y}_{(t-1)} = \tilde{w}\left[y_{(t-1)}; c_{(t-1)}\right] + \tilde{b} \tag{15}$$

where, $\tilde{w}$ and $\tilde{b}$ are related to the size of the decoder input, $\tilde{y}_{(t-1)}$ is the new decoder state. Through the LSTM unit, the decoder state $\tilde{y}_{(t-1)}$ can be converted into the final output, as shown in Equation (16),

$$d_t = g\left(d_{(t-1)}, \tilde{y}_{(t-1)}\right) \tag{16}$$

where, g is the LSTM unit, and d_t is the final output.

The training parameter settings of the multi-factor prediction model are shown in Table 2.

TABLE 2. The training parameter settings of the multi-factor prediction model.

Hyperparameter	Specific settings
Number of iterations	1000
Time Step	48
Learning rate	0.01
Loss function	L2 loss
Batch size	24
Input dimension	48×12
Output dimension	6×1
Encoder layers	2
Encoder neural unit per layer	64
Decoder layer number	2
Decoder neural unit of each layer	64

D. DUAL MODEL INTEGRATION USING XGBoost

The main reason we chose the XGBoost includes two aspects. One is the characteristics of air quality data, which is a type of data set with complex and various characteristics. The other is that XGBoost can integrate multiple types of models, which is consistent with our needs. XGBoost improves the accuracy of prediction results by adding trees one by one. In XGBoosting tree, each addition of subtrees will improve accuracy of the prediction results, and finally the most accurate prediction results are obtained.

XGBoost integrates multiple trees into a strong classifier. The feature with the largest information gain is selected as the split point. The feature is split by continuously adding trees. Each leaf node of the subtree is assigned a weight value. In each round of training, the weight value of each leaf node is adjusted according to the objective function. After multiple iterations, the optimal subtree is achieved. Each leaf node in the optimal subtree corresponds to a predicted value.

The CART (Classification and Regression Tree) is the base learner of XGBoost. But the output results of CART tree and XGBoosting tree are generated with different ways. Each output value of XGBoosting tree is a weighted sum generated by a function, while each output result in the CART regression tree is the mean value of all sample points of the leaf node.

The input data of the integration process include the results of the single-factor model, the results of the multi-factor model and current weather data. Specifically, the input data of XGBoost regression are $x_i = \{p_i, z_i, g\}$, where p are the results of the single-factor model, z are the results of the multi-factor model, and g are the meteorological data at current moment. Meteorological data are used to flexibly adjust the weight of the air quality data to obtain accurate prediction values. For a given air quality data sample $\{x_i, y_i\}_{i=1}^{n}\}$, the process of completing the sequence prediction with XGBoost regression is as follows.

If there are k trees in XGBoost as in Equation (17),

$$\hat{y}_i = \sum_{i=1}^{k} f_k(x_i), \quad f_k \epsilon F \tag{17}$$

where, F represents all the function space in the regression forecast, $\hat{y}_i$ is the predicted value of the model, and $f_k(x_i)$ represents the weight value of the i^{th} sample in the k^{th} tree leaf. The structure of each tree and the weight of each leaf, f_k of each subtree, are what needs to be solved during the training process.

The structure of each tree and the weight of each leaf need to be solved by the objective function obj(Θ). For the parameter $\{\Theta = f_1, f_2, \ldots f_k\}$, the objective function is composed of an error function and a penalty function, and its formula is as Equation (18),

$$obj(\Theta) = L(\Theta) + \Omega(\Theta) \tag{18}$$

where, $L(\Theta)$ is the error function used to express the differences between the fitted value and the actual data, and $\Omega(\Theta)$ is the regularization term used to punish complex models.

During the training process, each tree will accumulate until it stops at k trees. The training process is shown in Equation (19),

$$\begin{aligned}
\hat{y}_i^{(0)} &= 0 \\
\hat{y}_i^{(1)} &= \hat{y}_i^{(0)} + f_1(x_i) = f_1(x_i) \\
\hat{y}_i^{(2)} &= \hat{y}_i^{(1)} + f_2(x_i) = f_1(x_i) + f_2(x_i) \\
&\vdots \\
\hat{y}_i^{(t)} &= \hat{y}_i^{(t-1)} + f_t(x_i) = \sum_{k=1}^{t} f_k(x_i)
\end{aligned} \tag{19}$$

where, t represents the t^{th} round of model training, and $\hat{y}_i^{(t)}$ represents the predicted value of x_i after the i^{th} round. The error during training is shown in Equation (20),

$$L(\Theta) = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i\right) = \sum_{i=1}^{n} l\left(y_i, \sum_{k=1}^{t} f_k(x_i)\right) \tag{20}$$

where, $l()$ represents the loss function which commonly is square loss or logistic loss, y_i is the labeled data, and $\hat{y}_i$ is the predicted data.

Each subtree of the XGBoost tree is established as Equation (21). Data features are mapped to the leaf nodes of each subtree.

$$f_t(x) = w_{q(x)}, w \epsilon R^T, q : R^d \rightarrow \{1, 2, \ldots, T\} \quad (21)$$

where, $f_t(x)$ represents the node prediction value of each subtree, w represents the weight value of the leaf, q represents the structure of the tree, T represents the number of leaves of the tree, and R_d represents the data set where the number of features is d.

The objective function can be used to evaluate the model, but it cannot avoid the phenomenon of overfitting. In order to reduce the model bias caused by data noise, penalty terms are added during training. The penalty term during training is shown in Equation (22),

$$\Omega(\Theta) = \sum_{k=1}^{t} \Omega(f_k) \quad (22)$$

After training, the subtree will be updated according to the optimal value of the objective function. The leaf node of each subtree, which represents a predicted value, will be adjusted. The prediction results of air quality can be obtained finally by accumulating the predicted value of each subtree.

We have implemented the XGBoost regression using Python and XGBoost library. The parameters and their training settings of the regression are shown in Table 3.

TABLE 3. The training parameter settings of the XGBoost regression.

Parameter	Value
Max_depth	5
Learning_rate	0.01
N_estimators	3602
Min_child_weight	3
Subsample	0.72
Scale_pos_weight	1
Silent	false
Gamma	0.1
Closample_bytree	0.75
Reg_alpha	0.00001

Where, *Subsample* is used to control the proportion of each tree randomly adopted. When its value is set to a smaller value, the regression may be more conservative and may reduce overfitting. *Gamma* represents the minimum loss function drop value required for node splitting. The larger the value is, the more conservative the model is.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

We have implemented detailed experiments to verify our proposed method. The experimental results show that our method has higher prediction accuracy than other methods.

A. EXPERIMENTAL ENVIRONMENT

We adopted the GPU version of TensorFlow as our experimental environment. Some other development tools, such as python, numpy, scikit-learn and XGBoost library, are used in our experiments. The detailed hardware configurations and software versions are shown in Table 4.

TABLE 4. The detailed hardware configurations and software versions.

Items	Description
CPU	Intel Xeon E5
GPU	Nvidia GTX970
RAM	32G
Hard disk	500G
Operating system	Ubuntu16.04
Development language	Python3
Development tools	Pycharm2017
Deep learning framework	Tensorflow1.8_GPU
Other libraries	Keras、XGBoost、Scikit-learn、Numpy、Scipy

B. EXAMPLE OF PREDICTION RESULTS

Based on our proposed prediction method, the comparison between the prediction results and the actual values for PM2.5 data are shown in Fig. 6. It can be seen from the figure that the method can obtain prediction results that are very similar to the actual values.

At the same time, for the PM10 data, the comparison between the prediction results and the actual values are shown in Fig 7, which can draw an approximate conclusion that the proposed method has a satisfying prediction accuracy.

C. EVALUATION METHOD OF EXPERIMENTAL RESULTS

Five evaluation methods, RMSE (Root Mean Square Error), MAE (Mean Absolute Error), MAPE (Mean Absolute Percentage Error), R-square and IA (Index of Agreement), were used to evaluate and analyze the final prediction results of our proposed model.

RMSE is the square root of the ratio of the square of the deviation between the predicted value and the true value. It is more sensitive to outliers in the data. RMSE is used as an evaluation index, whose calculation method is shown in the Equation (23),

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2}{n}} \quad (23)$$

where, y_i represents the true value, $\hat{y}_i$ represents the predicted value, and n represents the number of true values. The smaller *RMSE* value is, the stronger the model's ability to fit experimental data is.

MAE is mainly to better reflect the actual situation of the predicted value error, whose calculation method is

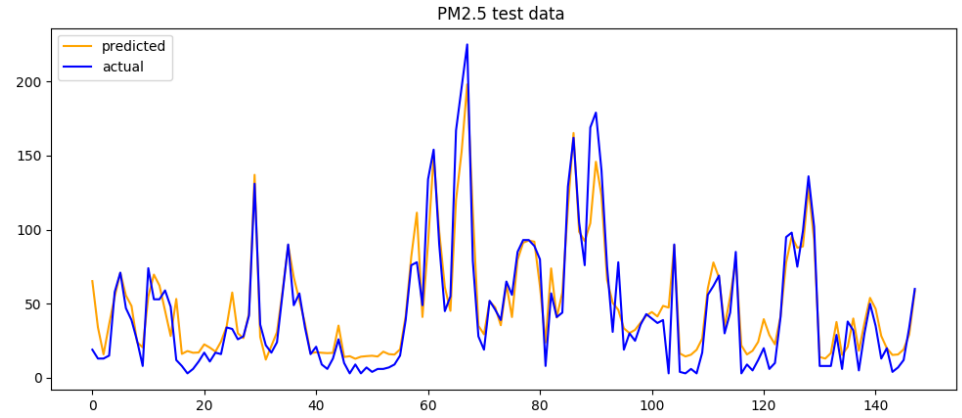

FIGURE 6. Comparison between the prediction results and the actual values for PM2.5 data.

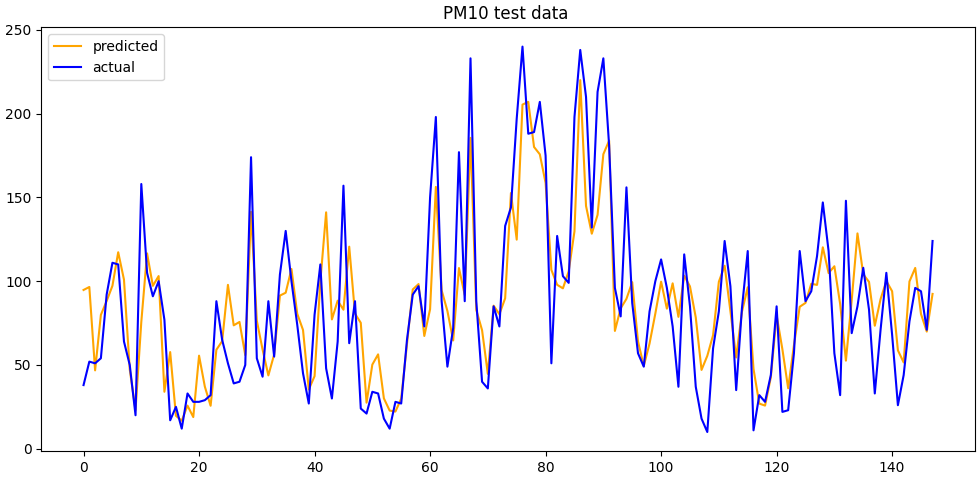

FIGURE 7. Comparison between the prediction results and the actual values for PM10 data.

shown in Equation (24),

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{24}$$

where, y_i represents the true value, $\hat{y}$ represents the predicted value, and n represents the number of true values.

MAPE is usually used to compare predictions of different proportions. Its calculation method is shown in Equation (25),

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \tag{25}$$

where, y_i represents the true value, $\hat{y}_i$ represents the predicted value, and n represents the number of true values. The advantage of MAPE is that it provides a benchmark for comparison. The lower the result of MAPE is, the better the model is. When the predicted $\hat{y}$ is exactly the same as the real y, the minimum of MAPE value is 0.

R-square is an important statistic reflecting the goodness of fit of the model. It is used as the evaluation index of regression analysis, whose calculation method is shown in Equation (26),

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \bar{y}_i\right)^2} \tag{26}$$

where, y_i represents the true value, $\hat{y}_i$ represents the predicted value, $\bar{y}_i$ represents the average value of y_i, and n represents the number of true values. The larger R-square is, the better the fitted regression model is.

IA is a dimensionless and bounded metric index which is commonly used to evaluate the average loss of the predicted values of the model. Its calculation method is shown in Equation (27),

$$IA = 1 - \frac{\sum_{i=1}^{n}\left(y_i - Y_i\right)^2}{\sum_{i=1}^{n}\left(\left|y_i - \bar{Y}_i\right| + \left|Y_i - \bar{Y}_i\right|\right)^2} \tag{27}$$

where, y_i is the true value, Y_i is the predicted value, $\bar{Y}_i$ represents the average value of Y_i. The bigger IA value is, the better the consistency of the model is.

TABLE 5. Comparison of RMSE error of multi-factor model under input and output sequence of various lengths.

x in Equation (5)	Lengths of Input Data	Output 1 Hour	Output 3 Hours	Output 6 Hours	Output 12 Hours	Output 18 Hours	Output 24 Hours	Output 48 Hours
0	24	25.98	33.28	39.41	43.18	44.03	44.19	45.00
2	36	29.63	37.43	43.15	43.94	44.26	44.49	44.98
4	48	18.77	26.42	31.69	36.83	40.41	43.10	47.69
8	72	19.94	30.91	39.01	43.31	44.47	45.00	45.77
16	120	41.14	39.43	41.96	45.38	46.90	47.93	49.53

TABLE 6. Comparison of the prediction error based on various models.

	RMSE	MAE	MAPE	R-square	IA
SVR	55.92	39.95	72.78	0.11	0.25
Ridge regression	79.23	45.85	86.96	0.51	0.24
XGBoost	68.24	20.21	90.97	0.78	0.37
SLSTM [31]	67.69	47.26	78.78	0.68	0.34
NLSTM [32]	47.90	37.08	73.43	0.65	0.24
Single-factor model	26.42	18.35	79.09	0.69	0.91
Multi-factor model	18.77	12.05	42.39	0.81	0.88
Our integrated model	14.36	8.39	35.78	0.89	0.93

D. PREDICTION ACCURACY CAUSED BY INPUT DATA OF VARIOUS LENGTHS

In Seq2Seq, the length of the input sequence will affect the expressive ability of the model, thereby affecting the prediction accuracy of the model.

Figure 8 shows the comparison of RMSE error of multi-factor model under input and output sequence of various lengths. The detailed data of Fig. 8 are descripted in Table 5. It can be seen from the table and figure that when the input sequence length is 48 (x = 4), the model will have the smallest error range.

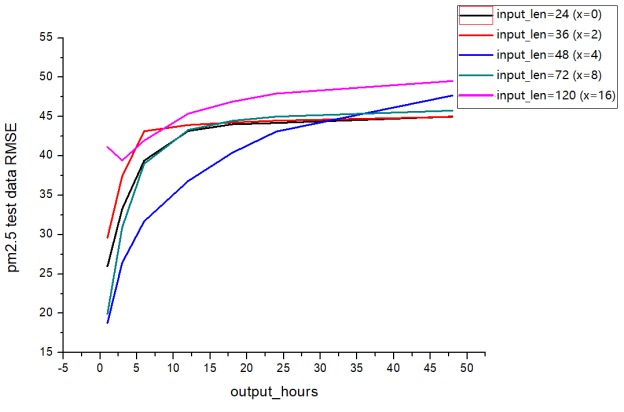

FIGURE 8. Comparison of RMSE error of multi-factor model under input and output sequence of various lengths.

E. COMPARISON BETWEEN OUR METHOD AND OTHER METHODS

We implemented our proposed integrated dual LSTM model method. Table 6 compares the error conditions of the prediction results of various models, including SVR (Support Vector Regression), Ridge Regression, pure XGBoost model, SLSTM [31], NLSTM [32], single-factor prediction model, multi-factor prediction model and our integrated dual LSTM model in this article.

Table 6 uses five evaluation indicators, RMSE, MAE, MAPE, R-square and IA, to evaluate the methods. The scores in the table are obtained when the length of the input sequences is 48 and the output is 1 hour. It can be seen from the table that our proposed prediction method can obtain less RMSE and MAE errors, and better performances in terms of error and expressiveness of the model.

V. CONCLUSION AND FUTURE WORK

In order to improve the accuracy of air quality data prediction, we proposed a prediction model based on integrated dual LSTM model method. The realization process and effect of the integrated model can be described as follows. Firstly, a single-factor prediction model is established to predict each component of air quality data. Then, a multi-factor prediction model is established to predict the data of the current station by combining the historical data of the current station and surrounding stations, as well as meteorological data. Next, XGBoost regression is adopted to build the optimal boost tree, in which the best prediction results can be obtained by combining the single-factor model and the multi-factor model.

The method in this paper combined two models, which were established in two dimensions of time and space, to obtain the best prediction results. First, the single-factor model for each factor was established in the time dimension. Single factors, like PM2.5, are used as the input of single-factor models. The forecast results are obtained

by using the characteristics in the time dimension. Then, the multi-factor model is established in the spatial dimension. Multiple factors, such as data from the current station and surrounding stations, as well as weather data, are selected as the input of the multi-factor model together. The prediction results are obtained according to the spatial characters. Finally, the XGBoosting tree takes the output of the single-factor time model and the multi-factor spatial model as input. The optimized predicted value is obtained by calculating the weight of each leaf node and accumulating its predicted value. Based on evaluating the experimental results using five evaluation indicators, the method proposed in this paper can obtain prediction results with higher accuracy.

In the future work, the next step of the research is to expand the range of application of the integrated dual LSTM model method to improve the accuracy of various data prediction due to the integration of the advantages of multiple models. In addition, we have found some prediction results with outlier values, although there are very small probabilities in the results of our model. The analysis of this kind of outlier value is one of the problems that need to be solved in the next step of this article.

ACKNOWLEDGMENT

The authors would like to thank the visualization laboratory of Beijing Technology and Business University for providing the location data set of air quality monitoring stations. They thank Wanlin Chen and Liping Sun for their contributions to the preliminary research of this article, thank Yuying Deng for providing us with detailed grammar revision work. They thank to the experts for their evaluation and suggestions on our method.

REFERENCES

[1] H. Petr and O. Vladimir, "Prediction of air quality indices by neural networks and fuzzy inference systems," *Commun. Comput. Inf. Sci.*, vol. 383, pp. 302–312, Sep. 2013, doi: 10.1007/978-3-642-41013-0_31.

[2] Z. Kang and Z. Qu, "Application of BP neural network optimized by genetic simulated annealing algorithm to prediction of air quality index in Lanzhou," in *Proc. IEEE Comput. Intell. Appl. (ICCIA)*, Sep. 2017, pp. 155–160, doi: 10.1109/CIAPP.2017.8167199.

[3] X. Wang and B. Wang, "Research on prediction of environmental aerosol and PM2.5 based on artificial neural network," *Neural Comput. Appl.*, vol. 31, no. 12, pp. 8217–8227, Dec. 2019, doi: 10.1007/s00521-018-3861-y.

[4] T. S. Rajput and N. Sharma, "Multivariate regression analysis of air quality index for Hyderabad city: Forecasting model with hourly frequency," *Int. J. Appl. Res.*, vol. 3, no. 8, pp. 443–447, 2017. Accessed: Mar. 20, 2021. [Online]. Available: https://www.allresearchjournal.com/archives/2017/vol3iss ue8/PartG/3-8-78-443.pdf

[5] S. Mahajan, H.-M. Liu, T.-C. Tsai, and L.-J. Chen, "Improving the accuracy and efficiency of PM2.5 forecast service using cluster-based hybrid neural network model," *IEEE Access*, vol. 6, pp. 19193–19204, 2018, doi: 10.1109/ACCESS.2018.2820164.

[6] R. Li, Y. Dong, Z. Zhu, C. Li, and H. Yang, "A dynamic evaluation framework for ambient air pollution monitoring," *Appl. Math. Model.*, vol. 65, pp. 52–71, Jan. 2019, doi: 10.1016/j.apm.2018.07.052.

[7] B. Liu, S. Yan, J. Li, G. Qu, Y. Li, J. Lang, and R. Gu, "A sequence-to-sequence air quality predictor based on the n-step recurrent prediction," *IEEE Access*, vol. 7, pp. 43331–43345, 2019, doi: 10.1109/ACCESS.2019.2908081.

[8] K. Gu, J. Qiao, and W. Lin, "Recurrent air quality predictor based on meteorology- and pollution-related factors," *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 3946–3955, Sep. 2018, doi: 10.1109/TII.2018.2793950.

[9] M. Benhaddi and J. Ouarzazi, "Multivariate time series forecasting with dilated residual convolutional neural networks for urban air quality prediction," *Arabian J. Sci. Eng.*, vol. 46, no. 4, pp. 3423–3442, Apr. 2021, doi: 10.1007/s13369-020-05109-x.

[10] X. Song, J. Huang, and D. Song, "Air quality prediction based on LSTM-Kalman model," in *Proc. IEEE 8th Joint Int. Inf. Technol. Artif. Intell. Conf. (ITAIC)*, Chongqing, China, May 2019, pp. 695–699, doi: 10.1109/ITAIC.2019.8785751.

[11] J. Wang, J. Li, X. Wang, J. Wang, and M. Huang, "Air quality prediction using CT-LSTM," *Neural Comput. Appl.*, vol. 33, no. 3, pp. 1–14, Nov. 2020, doi: 10.1007/s00521-020-05535-w.

[12] Z. Jianhui, D. Ting, and C. Bo, "AQI prediction based on long short-term memory model with spatio-temporal optimizations and fireworks algorithm," *J. Wuhan Univ.*, vol. 65, no. 3, pp. 250–262, 2019, doi: 10.14188/j.1671-8836.2019.03.004.

[13] D. Qin, J. Yu, G. Zou, R. Yong, Q. Zhao, and B. Zhang, "A novel combined prediction scheme based on CNN and LSTM for urban PM2.5 concentration," *IEEE Access*, vol. 7, pp. 20050–20059, 2019, doi: 10.1109/ACCESS.2019.2897028.

[14] S. Li, G. Xie, J. Ren, L. Guo, Y. Yang, and X. Xu, "Urban PM2.5 concentration prediction via attention-based CNN–LSTM," *Appl. Sci.*, vol. 10, no. 6, p. 1953, Mar. 2020, doi: 10.3390/app10061953.

[15] L. Zhang, P. Liu, L. Zhao, G. Wang, W. Zhang, and J. Liu, "Air quality predictions with a semi-supervised bidirectional LSTM neural network," *Atmos. Pollut. Res.*, vol. 12, no. 1, pp. 328–339, Jan. 2021, doi: 10.1016/j.apr.2020.09.003.

[16] S. V. Belavadi, S. Rajagopal, R. R, and R. Mohan, "Air quality forecasting using LSTM RNN and wireless sensor networks," *Procedia Comput. Sci.*, vol. 170, pp. 241–248, Jan. 2020, doi: 10.1016/j.procs.2020.03.036.

[17] X. Sun, W. Xu, and H. Jiang, "Spatio-temporal prediction of air quality based on recurrent neural networks," in *Proc. 52nd Hawaii Int. Conf. Syst. Sci. (HICSS)*, Jan. 2019, pp. 1–10, doi: 10.1016/j.aej.2020.12.009.

[18] X. Zou, J. Zhao, D. Zhao, B. Sun, Y. He, and S. Fuentes, "Air quality prediction based on a spatiotemporal attention mechanism," *Mobile Inf. Syst.*, vol. 2021, pp. 1–12, Feb. 2021, doi: 10.1155/2021/6630944.

[19] Z. Qin, C. Cen, and X. Guo, "Prediction of air quality based on KNN-LSTM," *J. Phys. Conf.*, vol. 1237, Jun. 2019, Art. no. 042030, doi: 10.1088/1742-6596/1237/4/042030.

[20] J. Zhao, F. Deng, Y. Cai, and J. Chen, "Long short-term memory—Fully connected (LSTM-FC) neural network for PM2.5 concentration prediction," *Chemosphere*, vol. 220, pp. 486–492, Apr. 2019, doi: 10.1016/j.chemosphere.2018.12.128.

[21] P.-W. Soh, J.-W. Chang, and J.-W. Huang, "Adaptive deep learning-based air quality prediction model using the most relevant spatial-temporal relations," *IEEE Access*, vol. 6, pp. 38186–38199, 2018, doi: 10.1109/ACCESS.2018.2849820.

[22] Y. Qi, Q. Li, H. Karimian, and D. Liu, "A hybrid model for spatiotemporal forecasting of PM2. 5 based on graph convolutional neural network and long short-term memory," *Sci. Total Environ.*, vol. 664, pp. 1–10, May 2019, doi: 10.1016/j.scitotenv.2019.01.333.

[23] D. Seng, Q. Zhang, X. Zhang, G. Chen, and X. Chen, "Spatiotemporal prediction of air quality based on LSTM neural network," *Alexandria Eng. J.*, vol. 60, no. 2, pp. 2021–2032, Apr. 2021, doi: 10.1016/j.aej.2020.12.009.

[24] Y. Zhou, F.-J. Chang, L.-C. Chang, I.-F. Kao, and Y.-S. Wang, "Explore a deep learning multi-output neural network for regional multi-step-ahead air quality forecasts," *J. Cleaner Prod.*, vol. 209, pp. 134–145, Feb. 2019, doi: 10.1016/j.jclepro.2018.10.243.

[25] R. Yan, J. Liao, J. Yang, W. Sun, M. Nong, and F. Li, "Multi-hour and multi-site air quality index forecasting in Beijing using CNN, LSTM, CNN-LSTM, and spatiotemporal clustering," *Expert Syst. Appl.*, vol. 169, no. 4, Dec. 2020, Art. no. 114513, doi: 10.1016/j.eswa.2020.114513.

[26] X. Xu and M. Yoneda, "Multitask air-quality prediction based on LSTM-autoencoder model," *IEEE Trans. Cybern.*, vol. 51, no. 5, pp. 2577–2586, May 2021, doi: 10.1109/TCYB.2019.2945999.

[27] L. Wang, Z. Zhu, L. Sassoubre, G. Yu, C. Liao, Q. Hu, and Y. Wang, "Improving the robustness of beach water quality modeling using an ensemble machine learning approach," *Sci. Total Environ.*, vol. 765, 2020, Art. no. 142760, doi: 10.1016/j.scitotenv.2020.142760.

[28] G. Huang, X. Li, B. Zhang, and J. Ren, "PM2.5 concentration forecasting at surface monitoring sites using GRU neural network based on empirical mode decomposition," *Sci. Total Environ.*, vol. 768, May 2021, Art. no. 144516, doi: 10.1016/j.scitotenv.2020.144516.

[29] Y.-S. Chang, S. Abimannan, H.-T. Chiao, C.-Y. Lin, and Y.-P. Huang, "An ensemble learning based hybrid model and framework for air pollution forecasting," *Environ. Sci. Pollut. Res.*, vol. 27, no. 30, pp. 38155–38168, Oct. 2020, doi: 10.1007/s11356-020-09855-1.
[30] J. Zhu, F. Deng, J. Zhao, and H. Zheng, "Attention-based parallel networks (APNet) for PM2.5 spatiotemporal prediction," *Sci. Total Environ.*, vol. 769, May 2021, Art. no. 145082, doi: 10.1016/j.scitotenv.2021.145082.
[31] Y. Bai, B. Zeng, and C. Li, "An ensemble long short-term memory neural network for hourly PM2.5 concentration forecasting," *Chemosphere*, vol. 222, pp. 286–294, May 2019, doi: 10.1016/j.chemosphere.2019.01.121.
[32] C.-C. Wei, "Development of stacked long short-term memory neural networks with numerical solutions for wind velocity predictions," *Adv. Meteorol.*, vol. 2020, no. 2, pp. 1–18, 2020, doi: 10.1155/2020/5462040.
[33] J. R. A. Moniz and D. Krueger, "Nested LSTMs," in *Proc. Asian Conf. Mach. Learn. (ACML)*, vol. 77, 2017, pp. 530–544.
[34] N. Jin, Y. Zeng, K. Yan, and Z. Ji, "Multivariate air quality forecasting with nested LSTM neural network," *IEEE Trans. Ind. Informat.*, early access, Mar. 17, 2021, doi: 10.1109/TII.2021.3065425.
[35] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794, doi: 10.1145/2939672.2939785.
[36] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc., B, Methodol.*, vol. 39, no. 1, pp. 1–22, Sep. 1977, doi: 10.1111/j.2517-6161.1977.tb01600.x.

HONGQIAN CHEN was born in Shandong, China, in 1982. He received the Ph.D. degree in computer application technology from the Beijing Institute of Technology, in 2009.

From 2009 to 2013, he was a Lecturer with the School of Computer Science and Engineering, Beijing Technology and Business University. Since 2013, he has been an Assistant Professor with the School of Computer Science and Engineering, Beijing Technology and Business University. He is the author of two books, more than 40 articles, and the inventor of more than 20 inventions. His research interests include computer vision, data mining, and data visualization.

MENGXI GUAN was born in Beijing, China, in 1997. She received the B.S. degree in software engineering from Beijing Technology and Business University, in 2019, where she is currently pursuing the master's degree with the School of Computer Science and Engineering.

Her research interests include data visualization and computer vision.

HUI LI was born in Shandong, China, in 1983. She received the Ph.D. degree in computer application technology from the University of Science and Technology, Beijing, in 2014.

From 2014 to 2018, she was a Lecturer with the College of Management, Beijing Union University. Since 2018, she has been an Assistant Professor with the College of Management, Beijing Union University. She is the author of one book, more than 20 articles, and the inventor of one invention. Her research interests include data analysis, recommended systems, and data mining.