



# Stacked ResNet-LSTM and CORAL model for multi-site air quality prediction

Xiangwei Cheng<sup>1</sup> · Wenwen Zhang<sup>2</sup> · Adrian Wenzel<sup>1</sup> · Jia Chen<sup>1</sup>

Received: 27 September 2021 / Accepted: 4 March 2022 / Published online: 11 April 2022  
© The Author(s) 2022

## Abstract

As the global economy is booming, and the industrialization and urbanization are being expedited, particulate matter 2.5 (PM<sub>2.5</sub>) turns out to be a major air pollutant jeopardizing public health. Numerous researchers are committed to employing various methods to address the problem of the nonlinear correlation between PM<sub>2.5</sub> concentration and several factors to achieve more effective forecasting. However, a considerable space remains for the improvement of forecasting accuracy, and the problem of missing air pollution data on certain target areas also needs to be solved. Our research work is divided into two parts. First, this study presents a novel stacked ResNet-LSTM model to enhance prediction accuracy for PM<sub>2.5</sub> concentration level forecast. As revealed from the experimental results, the proposed model outperforms other models such as boosting algorithms or general recurrent neural networks, and the advantage of feature extraction through residual network (ResNet) combined with a model stacking strategy is shown. Second, to solve the problem of insufficient air quality and meteorological data on some research areas, this study proposes the use of a correlation alignment (CORAL) method to carry out a prediction on the target area by aligning the second-order statistics between source area and target area. As indicated from the results, this model exhibits a considerable accuracy even in the absence of historical PM<sub>2.5</sub> data in the target forecast area.

**Keywords** PM<sub>2.5</sub> · ResNet · Ensemble learning · Correlation alignment · Sliding window

## 1 Introduction

The rapid development of the global economy over the past few years has mostly come at the expense of environmental pollution. As environmental pollution becomes increasingly serious, environmental governance and protection

gained rising attention from the public [1]. Inimical health impacts from exposure to outdoor air pollutants refer to complex interaction of pollutant compositions [2, 3]. Real-time air quality information is essential for air pollution inspection and strategic decisions making, which is helpful to protect human health from air pollutants [4]. Atmospheric particulate matter, as a crucial indicator of air quality, deserves to be monitored and predicted. Accurate PM<sub>2.5</sub> forecasting can significantly help in improving prompt and complete environmental quality information, thus allowing the government to take timely action for environmental protection [5]. Besides, the prediction model helps to investigate and study the complicated nonlinear relationship between PM<sub>2.5</sub> and meteorological factors [6].

Several research groups have conducted relevant prediction studies on PM<sub>2.5</sub> or PM<sub>10</sub> and other atmospheric pollutants that impair air quality [7]. Moreover, they have employed hidden semi-Markov model [8], support vector machine [9], neural networks [10, 11] and other prediction

✉ Wenwen Zhang  
wenwenzhang@usst.edu.cn

Jia Chen  
jia.chen@tum.de

Xiangwei Cheng  
xw.cheng@tum.de

Adrian Wenzel  
a.wenzel@tum.de

<sup>1</sup> Department of Electrical and Computer Engineering,  
Technical University of Munich, 80333 Munich, Germany

<sup>2</sup> College of Science, Institute of Deep-Sea Advanced  
Equipment System, University of Shanghai for Science and  
Technology, Shanghai 200093, China

models trained on historical data. These models have achieved a certain degree of forecasting capability, whereas for hourly forecasts the prediction accuracy must be further improvement. Since for many statistical models it is difficult to capture the complex nonlinear behavior of PM<sub>2.5</sub> concentration change with varying meteorological conditions, more sophisticated models need to be developed that extract features present in meteorological data at a given instant as well as temporal features to follow and consider the trend. In addition, some target locations lack historical data of air quality. However, classical supervised learning methods rely on a large amount of the past data and require observations, i.e. measured data from the target locations, to train the models [12]. To confront this problem, we propose to employ the method of correlation alignment-our second model-that partly overcomes the necessity of historical air quality data.

## 2 Related work

In comparison with the mature weather forecast, air quality prediction is affected by a variety of complex factors, including climate, traffic, topography, etc., which are extremely complex and nonlinear processes [13]. PM<sub>2.5</sub> concentration prediction methods are mainly divided into two categories: (1) Comprehensive regional scale model based on spatial geographic information and physical/chemical rules. (2) Data-oriented regression prediction model based on machine learning and deep learning algorithm (or models).

As Geographic Information System (GIS) [14], Remote Sensing (RS) [15] and Global Positioning System (GPS) [16] technologies are leaping forward, the PM<sub>2.5</sub> concentration at ground level can be estimated through remote sensing by measuring Aerosol Optical Depth (AOD) [17]. Researchers established a correlation between AOD data and (surface) PM<sub>2.5</sub> concentration to monitor the temporal and spatial distribution of PM<sub>2.5</sub> near the ground [18]. Ma et al. adopted satellite remote sensing to estimate ground-level PM<sub>2.5</sub> and developed a national-scale geographically weighted regression (GWR) model with fused satellite AOD as the principal predictor to forecast diurnal PM<sub>2.5</sub> concentration in China [19]. There are extremely complex physical and chemical reactions and gas-solid two-phase transformation processes between various air pollutants. Some comprehensive regional-scale models can simulate the emission, chemical transformation, and transport of air pollutants, such as NAQPMs, CAMx, and WRF-Chem [20]. Saide et al. proposed an air quality prediction system under the WRF-Chem model, capable of effectively estimating 24-h mean PM<sub>2.5</sub> and PM<sub>10</sub> concentrations for 2008 winter in Santiago, Chile [21]. Hong et al. developed a

statistical model to optimize the estimation of the initial conditions of aerosol in the WRF-Chem model and improved PM<sub>2.5</sub> forecasting by integrating data from Himawari-8 (a Japanese weather satellite) and ground observations. The experiments were conducted in parallel over eastern China, and results reveal that the model they proposed greatly improved the PM<sub>2.5</sub> predictions [22]. It is worth noting that the performance of this model is highly correlated with the amount of valid data. Comprehensive regional-scale models require satellite remote sensing to obtain continuous and dynamic data on a large scale. The management, analysis, and visualization of massive data greatly increase work costs. Besides, affected by differences in geographic information, this type of model can be only applied to a specific area.

PM<sub>2.5</sub> prediction pertains to time series problems. Classical time series forecasting approaches employ one-dimensional information and analyze changing trends of target value by complying with sequential variations. The representative autoregressive integrated moving average (ARIMA) model [23, 24] has been extensively employed to predict air pollutants concentration. However, the ARIMA model focuses only on historical air quality data, whereas the effect of meteorological factors cannot be considered comprehensively. For this reason, such a method commonly has limited accuracy in PM<sub>2.5</sub> prediction. As compared with general statistical models, applying machine learning algorithms can better address some complex nonlinear relationships between dependent variables, thereby achieving better accuracy in air quality prediction [25]. Zhu et al. proposed an hourly forecast of air pollution concentration (e.g., PM<sub>2.5</sub> and sulfur dioxide) with machine learning approaches. They used refined models to make a prediction based on the meteorological data of previous days instead of applying standard regression models [26]. Since the change of PM<sub>2.5</sub> concentration is affected by many variables, it shows strong nonlinear characteristics. Artificial neural networks have great advantages in solving complex nonlinear problems [27] and also are based on existing monitoring data for modeling, so they have become the research focus of many researchers in the field of air pollution prediction. Wang et al. utilized the Back Propagation (BP) neural network to forecast the PM<sub>2.5</sub> in the Fuling District of Chongqing and the predicted results of this model demonstrated a similar trend with the measured values [28]. Huang et al. proposed the deep learning model APNet based on convolutional neural network (CNN) and long short-term memory (LSTM), which uses the characteristics of CNN to automatically extract features and complete the layer-by-layer abstraction of multiple feature sequences of a single site. Compared with the BP network, the CNN and LSTM network structures and operations are more complicated. The authors showed that

this model is capable of predicting the PM<sub>2.5</sub> concentration in the next 1 h more accurately than using CNN or LSTM alone [29]. Yeo et al. proposed a deep learning model which combines a CNN and a gated recurrent unit (GRU) to forecast PM<sub>2.5</sub> concentrations at 25 stations in Seoul, South Korea. In comparison with LSTM, GRU is computationally more efficient and its performance is well-matched as well. According to the average index of agreement, their approach has greatly improved prediction accuracy on the target station [30].

The proposed model in this study is aimed to accurately predict the 6-h PM<sub>2.5</sub> concentration of multiple areas in Beijing and avoid the poor-fitting phenomenon of traditional model at high concentration points. A novel model is built based on ResNet for feature extraction, LSTM for time series analysis and model-stacking technique. Moreover, to address the missing air quality data and meteorological information in the target prediction area, a transfer learning model based on domain adaption is constructed, which maps the data from the source and target domains. The main contributions of this study are as follows:

1. This study builds a basic model termed as ResNet-LSTM [31], which uses a ResNet CNN to extract important information from the historical PM<sub>2.5</sub> concentrations and meteorological data. In the experiment, it is shown that after the feature extraction from ResNet, the prediction accuracy can be improved. In the low concentration range of PM<sub>2.5</sub> ( $\leq 75 \mu\text{g}/\text{m}^3$ ), ResNet-LSTM has a better fitting performance than general recurrent neural networks. The average root-mean-square error, mean absolute error and R-Squared of the ResNet-LSTM model are  $44.886 \mu\text{g}/\text{m}^3$ ,  $25.946 \mu\text{g}/\text{m}^3$  and 0.760, respectively.
2. To more effectively enhance the forecasting capability of the single model, an ensemble learning method termed as stacking is adopted to combine several basic models. As verified in the comparative experiments, the proposed stacked ResNet-LSTM model outperforms the single basic model when adopted to the test dataset, especially when the observed PM<sub>2.5</sub> values are at a high concentration ( $\geq 150 \mu\text{g}/\text{m}^3$ ). The average root-mean-square error ( $40.679 \mu\text{g}/\text{m}^3$ ), mean absolute error ( $23.746 \mu\text{g}/\text{m}^3$ ) and R-Squared (0.804) verify the feasibility of the stacked ResNet-LSTM model. This stacked model can be used to create an early warning system for high PM<sub>2.5</sub> concentrations.
3. To address the lack of historical air quality and meteorological data in some target prediction areas, this study also proposes a domain adaptation method termed as correlation alignment to achieve the PM<sub>2.5</sub> concentration forecasting on these areas. This method can be used as a supplement to the stacked ResNet-

LSTM model in the absence of a large amount of historical data at other areas of interest.

### 3 Data sources

The research city of this study is Beijing, which is the political and economic center of China. The original data used in this study consist of two types, air quality data and meteorological data. Hourly PM<sub>2.5</sub> concentrations ( $\mu\text{g}/\text{m}^3$ ) were collected from the Beijing Municipal Environment Monitoring Center (BMEMC). The air quality monitoring sites in the dataset are located in 10 different locations in Beijing. The location, latitude, and longitude of these stations are shown in Table 1. These air quality monitoring stations are distributed in various areas in the city center and suburbs of Beijing, as presented in Fig. 1.

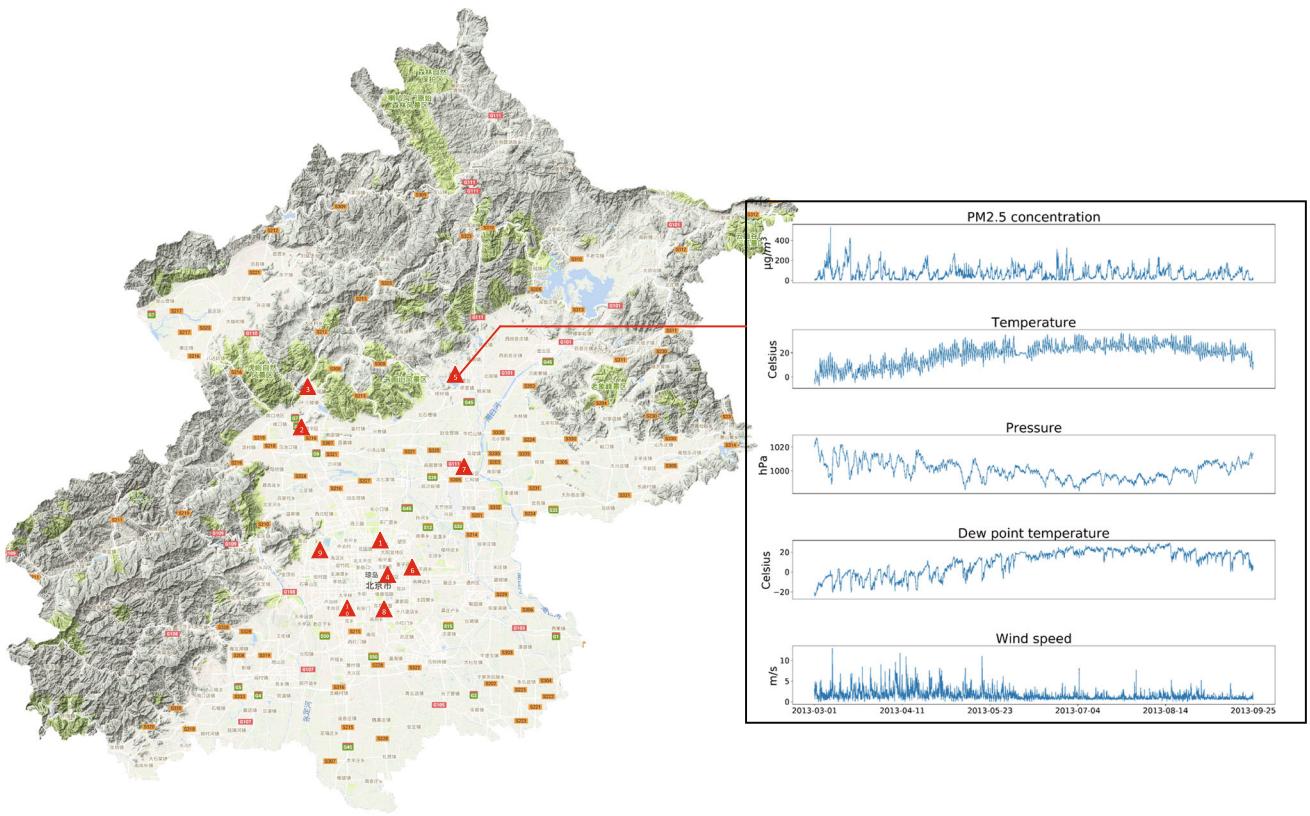
Original meteorological information including temperature ( $^\circ\text{C}$ ), pressure (hPa), dew point temperature ( $^\circ\text{C}$ ), wind direction and wind speed (m/s) were collected from observing stations of the China Meteorological Administration (CMA). The meteorological station is located close to the air monitoring stations in Beijing. On the whole, the combined historical PM<sub>2.5</sub> and meteorological dataset have over 35,000 h of samples from March 1, 2013, to February 28, 2017. Each monitoring point has a dataset of over 35,000 samples.

#### 3.1 Data pre-processing

Historical monitoring data collected from air quality monitoring stations and meteorological stations face several problems (e.g. missing data and inappropriate data formats). Thus, the original data are preprocessed according to the input requirements of the model for training samples.

**Table 1** Locations of air quality monitoring stations in Beijing

Number	Location	Latitude (N)	Longitude (E)
1	Aotizhongxin	39.990	116.404
2	Changping	40.230	116.233
3	Dingling	40.300	116.233
4	Dongsi	39.939	116.435
5	Huairou	40.308	116.647
6	Nongzhanguan	39.937	116.461
7	Shunyi	40.127	116.655
8	Tiantan	39.886	116.407
9	Wanliu	39.987	116.287
10	Wanshouxigong	39.878	116.352



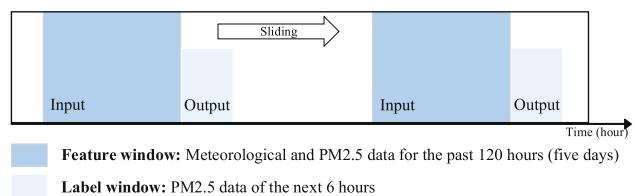
**Fig. 1** Geographical distribution map of Beijing Air Quality Monitoring Stations

The wind direction from original data is expressed in discrete string form, such as “east wind”, “southeast wind”, “northeast wind”. Here, we use a Label Encoding method for the numerical coding of the discrete feature. The 16 different wind directions is coded from 1 to 16. There are few missing data for features such as temperature, air pressure, wind direction, and wind speed (missing meteorological information ranging from 0.3 of to 0.8 % per station), and but those are not the main factors affecting the future changes of PM<sub>2.5</sub> concentration (missing PM<sub>2.5</sub> ranging from 1.1 of to 2.6 % per station). Accordingly, the “backfill” function in Pandas is employed to fill the missing data of these features. Based on the data information of other complete features, the missing PM<sub>2.5</sub> concentration value can be filled with a simple random forest model.

In total, the dataset provides 1461 days of hourly data for air quality and meteorological parameters. In this study, the meteorological data and PM<sub>2.5</sub> data of the first 5 days were chosen as the input feature of the neural network, and the PM<sub>2.5</sub> data of the first 6 h of the sixth day were treated as the corresponding labels, which corresponds to a forecast of 6 h into the future. However, through this method to separate the dataset, the final prediction result of the proposed model is not continuous. To increase the amount of

data and make the final prediction output result continuous, a method termed as sliding window was used in this study. By using this mechanism, a high-dimension temporal feature can be constructed for improving the prediction accuracy. Figure 2 illustrates the sliding window method.

The feature window consists of the historical meteorological data and air quality data from the past 120 h. The label window includes the following 6 h PM<sub>2.5</sub> concentration data. Every moving step of this mechanism is 6 h, which ensures that the PM<sub>2.5</sub> concentration value for a complete day can be predicted. After the data are processed by using the sliding window method, a total of about 5800 samples are obtained. In the model training process, 69% of the samples in the dataset are employed for training, 17% for validation, as well as 14% for testing. Table 2 lists the distribution of the complete dataset.



**Fig. 2** Sliding window method

**Table 2** The distribution of the complete dataset

Dataset	Percentage (%)	Amount
Training set	69	4000
Validation set	17	1000
Test set	14	824

## 4 Methods

Based on air quality data and meteorological information in Beijing, this study proposes a data-driven ensemble learning model termed stacked ResNet-LSTM. The PM<sub>2.5</sub> prediction model combines a convolutional neural network (ResNet) that learns local abstract features and a recurrent neural network (LSTM) with long-term memory function, to extract the temporal features of Beijing air quality and weather data. On that basis, an ensemble learning method is adopted to integrate multiple basic models to further enhance the prediction performance of the model. This happens through a stacking method of the separately trained basic models.

To address the missing historical air quality and meteorological data in some target prediction areas, this study also proposes to utilize a domain adaptation method termed as correlation alignment (CORAL) to achieve a forecast of PM<sub>2.5</sub> concentration on different target areas. The entire modeling and training process of the stacked ResNet-LSTM model and the CORAL model is shown in Fig. 3.

### 4.1 Stacked ResNet-LSTM model

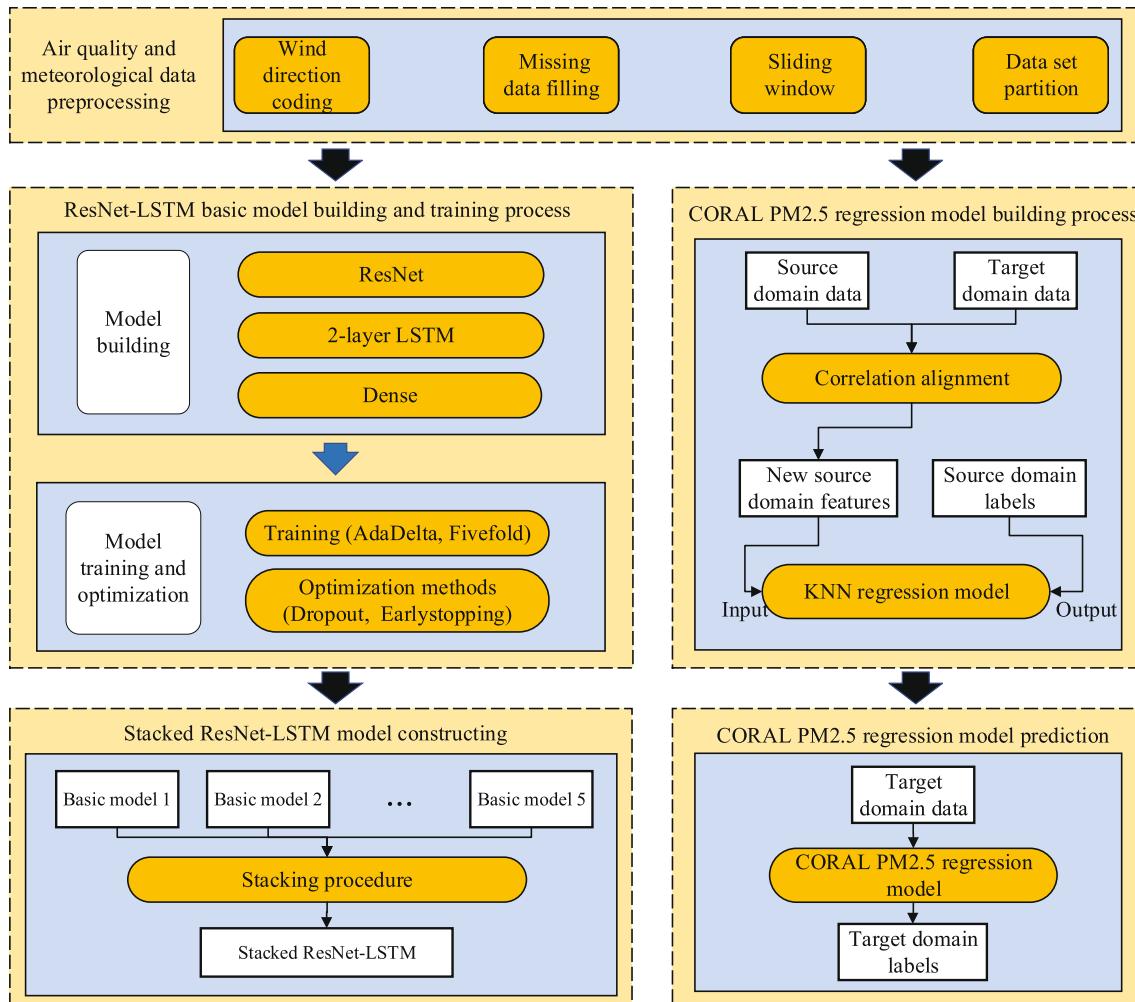
Before integrating several basic models by using an ensemble learning strategy, a basic model termed as ResNet-LSTM needs to be constructed. The architecture of our ResNet-LSTM basic model is shown in Fig. 4. The input of the ResNet-LSTM model is the data of the PM<sub>2.5</sub> concentration, temperature, pressure, dew point temperature, wind speed, and wind direction over the last five days, a total of 120 h as described in Sect. 3.1. The corresponding labels are the PM<sub>2.5</sub> concentration of the following 6 h. The proposed model can be divided into three main parts, namely ResNet for local feature extraction in PM<sub>2.5</sub> and meteorological data, LSTM for temporal dimension extracted features analysis, and fully connected layers for final regression prediction of PM<sub>2.5</sub> concentrations. The basic prediction model ResNet-LSTM is a 37-layer deep network and consists of the following three parts:

- (1) *ResNet* ResNet as a classical convolutional neural network is formulated by a series of basic blocks

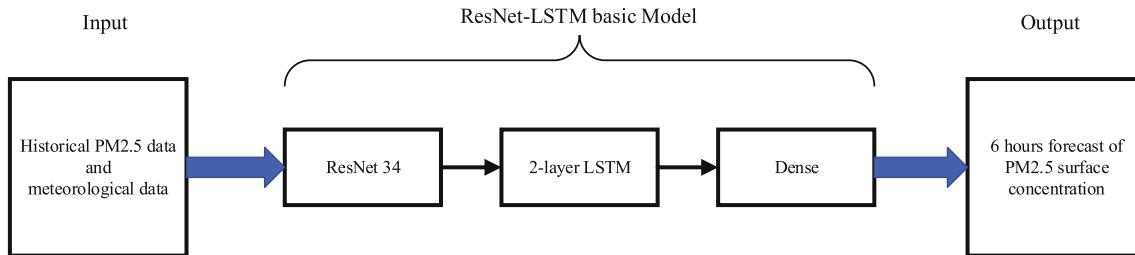
called residual blocks which can learn the desired mapping by utilizing a special' shortcut connection' architecture [32]. As shown in Fig. 5, a desired underlying mapping  $H(x)$  is fitted by two stacked nonlinear layers while the skip connection or identity mapping is implemented in the basic block to skip the transformation  $F(x)$  to form residual learning. As an alternative to learning a mapping of  $H(x)$ , the network learns a residual mapping based on the residual function  $F(x)$ . The ResNet model employed in this study is ResNet-34. Its structure of ResNet-34 is shown in Fig. 6. Since the fully connected layer is removed, it is a 33-layer convolutional neural network.

- (2) *LSTMs* LSTM refers to an optimized version of recurrent neural network, suitable for processing and predicting important events with relatively long intervals and delays in time series [33]. Our basic model comprises 2 LSTM layers, building a stacked LSTM architecture. The configuration of a LSTM cell consists of three parameters: *batch\_size*, *time\_step* and *input\_size*. The batch size of LSTM is identical to that of the entire model, which is set to 50 in advance. In the stacked LSTM part of the model, the time step of LSTM is set to 1, indicating that the length of the characteristic sequence processed at one time is 1. The parameter *input\_size* is set to 64, which means the length of a set of feature sequences is 64. In the 2-layer LSTM, each LSTM layer outputs a sequence of vectors used as an input to a subsequent LSTM layer. This hierarchy of hidden layers enables a more complex representation of our time-series data, so information at different scales is captured.
- (3) *Dense* The third part of our model consists of two fully connected (dense) layers. Based on a linear activation function, the basic model outputs the predicted PM<sub>2.5</sub> concentration on the target Beijing monitoring station area for the next 6 h.

The dimensional transformation of input features in the proposed network is presented in Fig. 7. After data pre-processing, the input is reshaped into a  $120 \times 6 \times 1$  matrix. The matrix shape originates from 120 h (5 days) of input data with 6 features used (PM<sub>2.5</sub>, temperature, pressure, dew point temperature, wind speed, and wind direction); furthermore, the matrix is expanded into the third dimension to be suitable for convolutional operations. Passing through the ResNet CNN, the shape of the input matrix is changed to  $15 \times 1 \times 512$ . The subsequent LSTM layers extract and process temporal features. Lastly, the data are sent to multiple fully connected layers to generate



**Fig. 3** Flowchart of the Stacked ResNet-LSTM model and the CORAL model

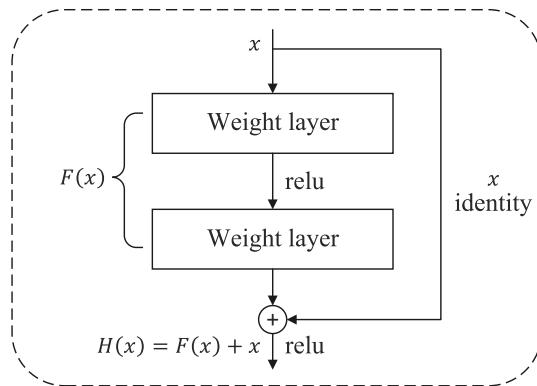
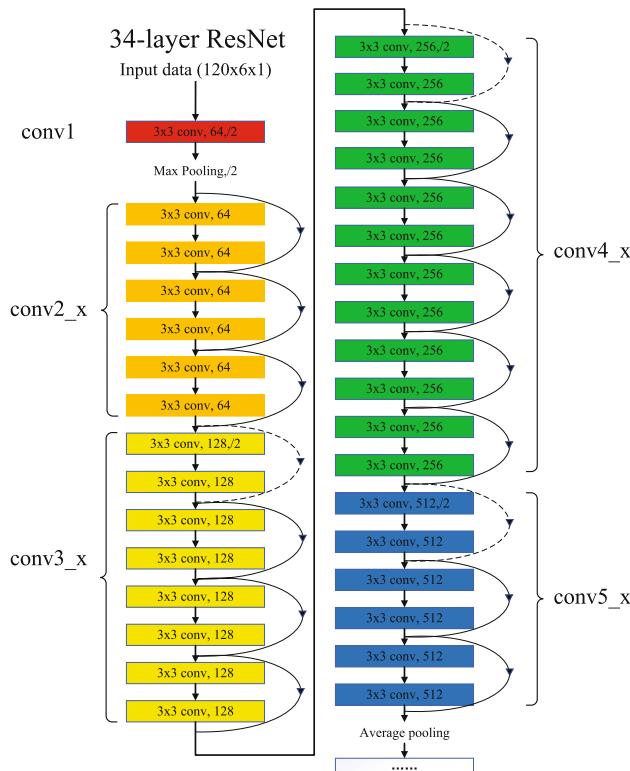


**Fig. 4** ResNet-LSTM model structure

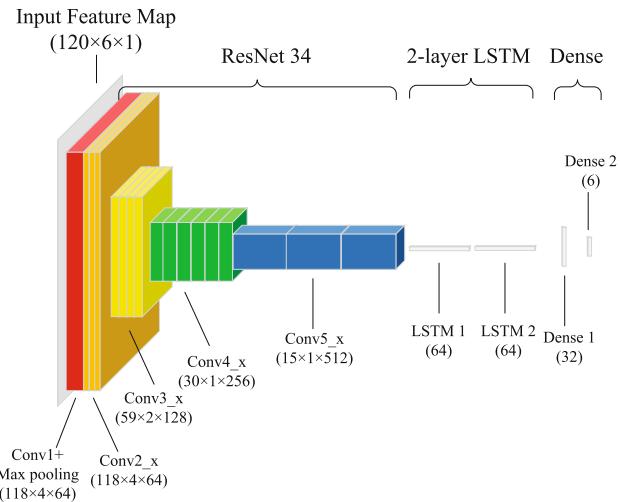
forecasting results. The successive layers and the corresponding output shape are listed in Table 3.

We employ the mean squared error (MSE) as loss function and optimize the model by adopting AdaDelta optimizer with a learning rate of  $1e-4$  [34]. The basic model is trained for 300 epochs, and a batch size of 50 is used. For different monitoring stations, the corresponding pre-processed data are adopted to train the basic models. To expedite the convergence of the model and reduce

overfitting, several optimization methods (e.g. Dropout, Regularization and EarlyStopping) are adopted. The operation of dropout is to randomly set some neurons in the hidden layer to 0 in accordance with a certain ratio. By exploiting the dropout strategy, the model is forced to learn more robust features and lower the impact of noise. Dropout was applied to the first layer and second layer of the stacked LSTMs part to enable its ability to generalize.

**Fig. 5** Residual block [32]**Fig. 6** ResNet-34 [32]

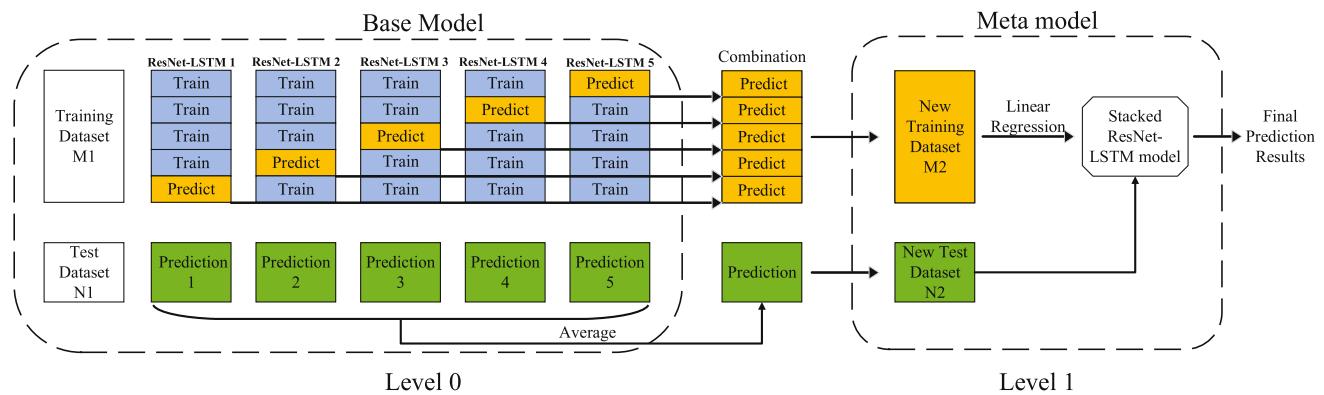
Moreover, the EarlyStopping method in the model is to prevent overfitting and expedite the completion of training. Before the experiments, we need to decide when to stop training according to the current validation result. For instance a decrease in training loss, but increase in validation loss suggests the beginning of overfitting. In this scenario, the training will be stopped automatically. The parameters of Earlystopping are set to `patience = 30`, `restore_best_weights = True`, which tells how many epochs can be tolerated without model improvement (= decrease in training and validation loss) and that the weight values of the model in its optimal state will be saved.

**Fig. 7** Dimension transformation in ResNet-LSTM model**Table 3** Successive layers and output shape of our suggested model ResNet-LSTM

Layer	Output	Filter	Stride
conv1 + max pooling	$118 \times 4 \times 64$	$3 \times 3 @ 64$	1
conv2_x	$118 \times 4 \times 64$	$3 \times 3 @ 64$	1
conv3_x	$59 \times 2 \times 128$	$3 \times 3 @ 128$	2
conv4_x	$30 \times 1 \times 256$	$3 \times 3 @ 256$	2
conv5_x	$15 \times 1 \times 512$	$3 \times 3 @ 512$	2
avg pooling	512		
LSTM 1	64		
LSTM 2	64		
Dense 1	32		
Dense 2	6		

A 5-fold validation method was adopted to train each basic model with its corresponding dataset. Then, a meta-model (stacked ResNet-LSTM) was constructed through a stacked generalization strategy. Stacked generalization was firstly proposed by David H. Wolpert in 1992 [35], which is an ensemble method that utilizes a high-level model to integrate several base models to achieve better performance in different machine learning tasks such as classification and regression.  $k$ -fold cross-validation is used on each base model to avoid the occurrence of overfitting and the ability of the model to generalize.

Considering the expected performances of the basic models and weighting the contribution of each sub-model to the combined prediction can improve the average performance of the model significantly. The modeling process of the stacked ResNet-LSTM model is demonstrated in Fig. 8.



**Fig. 8** ResNet-LSTM models stacking process

Each single ResNet-LSTM model is treated as a sub-model, and the linear regression acts as a meta-learning machine to construct a stacked ResNet-LSTM model. The specific implementation is as follows:

- Step 1. First split the original samples into training dataset M1 and test dataset N1.
- Step 2. Train each level 0 base model on the training dataset and test them on the test dataset (5-fold cross-validation is utilized on each base model).
- Step 3. Use predictions of validation dataset as inputs and corresponding target values to form a new training dataset M2 which is used to train the level 1 meta-model (linear regression here).
- Step 4. Average the predictions of the test dataset N1 to compose a new test dataset N2, and make a final prediction on the meta-model.

## 4.2 CORAL PM<sub>2.5</sub> regression model

In order to address the data shortage problem of historical air quality data in the target prediction area, our study adopts a transfer learning method. By aligning the second-order statistics between source domain (auxiliary prediction area) and target domain (target prediction area), a novel transfer learning method termed as CORrelation ALignment (CORAL) is adopted to develop an accurate and effective forecasting model that can be transferred to another prediction area.

### 4.2.1 Correlation alignment

CORAL refers to a classic statistical transfer learning method by complying with statistical feature transformation. The basic principle of the statistical feature alignment method is to transform and align the second-order statistics (covariance) between source domain and target domain [36]. The aligned covariances can be learned by adopting

traditional machine learning methods to develop a classifier. The main application of the CORAL model is image classification. In the proposed model, however, after the source domain and target domain data are aligned, a KNN regression algorithm can be adopted to predict the PM<sub>2.5</sub> concentrations on the target domain.

Suppose a labeled source domain dataset  $D_s = \{x_i\}_{i=1}^{n_s}$ ,  $x \in R^d$  with labels  $L_s = \{y_i\}_{i=1}^{n_s}$ , and an unlabeled target domain dataset  $D_t = \{u_j\}_{j=1}^{n_t}$ ,  $u \in R^d$ , where  $n_s$  and  $n_t$  are the number of samples in source domain and target domain respectively. Both  $x$  and  $u$  are the d-dimensional feature representations. The correlation alignment method aligns the two domains with second-order features. Assuming that  $C_{rms}$  and  $C_{rmt}$  are the covariance matrices of the source domain and the target domain, it learns a second-order feature transformation  $A$  that minimizes the feature distance between the source domain and the target domain.

$$\min_A \|C_s - C_t\|_F^2 = \min_A \|A^T C_s A - C_t\|_F^2, \quad (1)$$

where  $C_s$  represents the covariance matrix after the correlation alignment,  $A$  is the matrix for linear transformation.  $\|\cdot\|_F^2$  represents the matrix Frobenius norm.

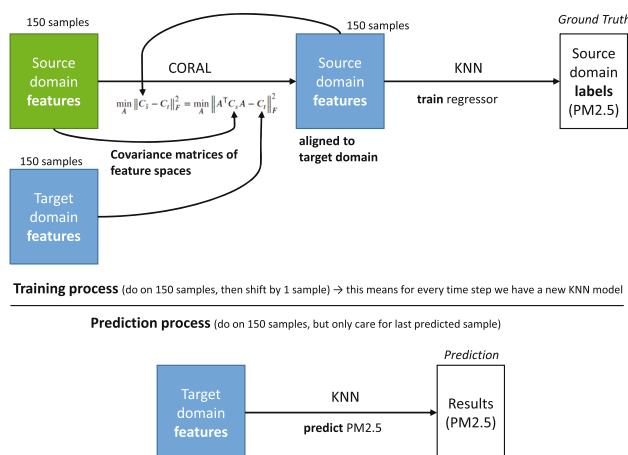
By solving the above optimization objective, the transformation matrix  $A$  can be obtained. The specific derivation process is explained in the “Appendix”.

### 4.2.2 K-Nearest Neighbor regression

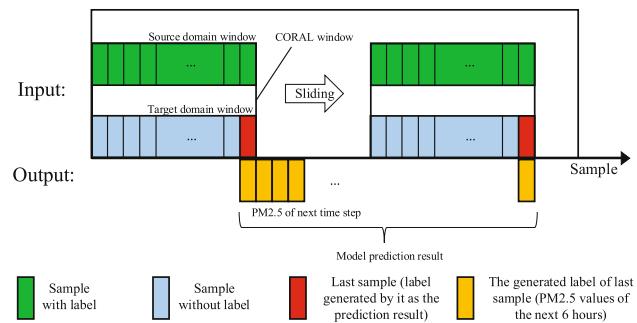
The K-Nearest Neighbor (KNN) regression algorithm refers to an example-based learning method, with the core idea to build a vector space model. By exploiting a certain distance measurement method, this algorithm aims at finding the neighbor points in the training set that are the nearest to the test point, and then employing the neighbor points to predict the test set. The average output of the neighboring points acts as the prediction result.

After the optimal  $A$  linear transformation matrix is generated by using the CORAL method,  $A$  can be utilized with the original source domain data to generate new source domain features. The KNN algorithm is then employed to build a regression model according to the procedure mentioned before. The input of this regression model is the newly generated (or transformed) features of the source domain, and the corresponding labels refer to the original labels from source domain. Then, the target domain features are given into the KNN regression model to obtain the corresponding labels to the target domain, which is the actual regression prediction result.

The training and prediction process of the CORAL PM<sub>2.5</sub> model is shown in Fig. 9. Here, we use 150 source domain samples and 150 target domain samples for training. 120 h of PM<sub>2.5</sub> concentration, temperature, pressure, and other features are also used as the input of one sample and the corresponding label in the source domain is the PM<sub>2.5</sub> concentration of the following 6 h. A sliding window mechanism is also utilized for the training and prediction of the CORAL PM<sub>2.5</sub> regression model. As shown in Fig. 10, a sliding window named CORAL window is composed of a source domain window and a target domain window. A source domain window and a target domain window include 150 samples respectively as mentioned before. After aligning the covariances between the input of the source domain and target domain, a KNN regression model is trained based on the transformed features of the source domain and their corresponding labels, which is then applied to the target domain to generate prediction results. Keeping the result of the last sample, the sliding window moves forward by one sample.



**Fig. 9** Idea of CORAL PM<sub>2.5</sub> regression model



**Fig. 10** Sliding window of CORAL PM<sub>2.5</sub> regression model

## 5 Simulation analysis

### 5.1 Evaluation metrics

Three evaluation metrics are adopted to evaluate the experimental results, which are root-mean-squared error (RMSE), mean absolute error (MAE), as well as R-Squared ( $R^2$ ). The RMSE is the arithmetic square root of the mean squared error, which is employed to measure the deviation between the real value and the predicted value. The MAE indicates the average absolute error between the observed value and predicted value. Smaller RMSE and MAE represent better model performance. The values of R-squared ranges from 0 to 1; the closer to 1, the better the fitting results of the model will be. The formulas of the three metrics are written as follows:

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

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (3)$$

$$R^2 = \frac{\sum_i (\hat{y}_i - \bar{y})^2}{\sum_i (y_i - \bar{y})^2}, \quad (4)$$

where  $y_i$ ,  $\hat{y}_i$  and  $\bar{y}$  represent true value, prediction value and mean value.

### 5.2 Methods for comparison

The proposed models are compared with the following methods:

- (1) *Gradient Boosted Decision Tree (GBDT)* GBDT is a gradient boosting of weak learners using CART (Classification and Regression Tree) [37]. One characteristic of using a decision tree as a weak learner is that the decision tree itself is an unstable learner, indicating that a slight fluctuation of the training data may significantly impact the results. From a statistical perspective, the variance of a

single decision tree is relatively large. In ensemble learning, however, the larger the variance between the weak learners, the better the generalization performance of the weak learners, and ultimately of the entire ensemble learning model.

- (2) *LightGBM* It refers to Microsoft's open-source distributed high-performance gradient boosting framework, which applies a learning algorithm based on decision trees [38]. LightGBM adopts a histogram-based decision tree algorithm to discretize continuous floating-point eigenvalues into  $k$  integers and construct a histogram with a width of  $k$ . It abandons the level-wise decision tree growth strategy followed by GBDT and uses a leaf-wise strategy with depth restrictions, which is a more efficient tree growth strategy compared to level-wise. On each step, it identifies the leaf with the largest split gain among all current leaves, then splits. Accordingly, compared with level-wise, leaf-wise is capable of reducing more errors and achieving better accuracy in the identical number of splits.
- (3) *Long short-term memory (LSTM)* As a type of recurrent neural network, it is also capable of learning order dependence in time-series prediction problems [39].
- (4) *Gated recurrent unit (GRU)* GRU is a further developed version of the standard recurrent neural network, i.e., a variant of LSTM. As compared with LSTM, the construction of GRU is simpler since it is reduced by one gate, and matrix multiplication is less strained. GRU involves fewer parameters than LSTM, so it exhibits a faster training speed and requires fewer samples. Nevertheless, if training data is enough, the test performance of LSTM may be better than GRU because LSTM has more flexibility when it comes to writing, reading and flushing the cell. As a variant of LSTM, GRU combines the input gate and the forget gate in LSTM into one, termed as “Update Gate”, which controls the amount of data that the previous memory information continues to retain to the current moment. Further, as there is no hidden memory state  $M$  in the GRU as opposed to the LSTM, the GRU's update gate additionally acts as the output gate in the LSTM [40].
- (5) *ResNet-LSTM* As a basic model, ResNet-LSTM can also be used in time series forecasting.
- (6) *Transfer component analysis (TCA)* As a kind of transfer learning algorithm, TCA is proposed to solve domain adaption problems by learning a shared subspace by minimizing the dissimilarities across domains [41]. (1), (2), (3), (4), (5) will act as the comparison algorithms in comparative experiments with stacked ResNet-LSTM, and (5), (6) will be used

in comparative experiments with transfer learning model CORAL.

In the comparative experiments, 300 iterations are performed for the respective neural network model and utilized the stochastic gradient descent technique for training. Moreover, the number of batches is set to 50, the value of learning rate is  $1e-4$ , the number of LSTM/GRU units (dimensionality of the output space) is set to 64. Specific to boosting algorithms and TCA, the optimal parameters of the mentioned models are found with the grid search method. The specific parameter settings are listed in Table 4.

### 5.3 Result analysis

#### 5.3.1 Comparative experiments of stacked ResNet-LSTM

Tables 5, 6 and 7 list the three evaluation metrics of the stacked ResNet-LSTM model and its comparative models at 10 air quality monitoring stations in Beijing.

The proposed model stacked ResNet-LSTM shows its prominent performance in accordance with the three evaluation indexes. As is shown in Fig. 11, the average RMSE (40.679), MAE (23.746) generated by the proposed model are the lowest among all the models and its  $R^2$  (0.804) is also the highest. It is therefore demonstrated that the proposed model stacked ResNet-LSTM exhibits significantly better universality in different target prediction areas to achieve high accuracy forecasting as compared with other algorithms. Since ResNet exhibits an outstanding ability of feature extraction and stacking generalization could further enhance forecasting ability, the stacked ResNet-LSTM model has thus better prediction performance compared with boosting algorithms and general neural network models. As indicated in Fig. 11, the classic recurrent neural network LSTM has a slightly better

**Table 4** Model parameter settings

Model	Key parameters
GBDT	max_depth=4, n_estimators=500, random_state=rng, learning_rate=0.01, loss='ls'
LightGBM	objective='regression', num_leaves=21, n_estimators=200, learning_rate=0.05
LSTM	batch_size=50, epochs=300,
GRU	learning_rate=1e-4, hidden_units=64,
ResNet-LSTM	dropout=0.3, optimizer=AdaDelta
TCA	kernel_type='primal', dim=30, lamb=1, gamma=1

**Table 5** The experimental results in terms of RMSE

Model	1	2	3	4	5	6	7	8	9	10	Average
GBDT	47.461	40.582	38.680	52.358	38.571	50.921	48.130	45.478	44.005	52.169	45.835
LightGBM	46.075	40.202	39.487	51.329	39.047	50.301	47.479	45.801	43.913	51.630	45.526
LSTM	48.985	46.212	40.790	55.904	42.512	52.001	54.148	48.250	47.552	52.375	48.873
GRU	48.530	44.051	40.640	58.101	44.202	51.038	54.675	48.470	49.748	57.943	49.740
ResNet-LSTM	46.321	41.454	37.977	52.680	38.994	48.686	46.561	43.925	42.920	49.346	44.886
Stacked	41.831	36.040	34.404	47.745	34.552	44.563	41.755	41.077	39.246	45.582	40.679
ResNet-LSTM											

**Table 6** The experimental results in terms of MAE

Model	1	2	3	4	5	6	7	8	9	10	Average
GBDT	26.101	22.559	20.609	29.264	20.523	28.020	27.304	25.627	24.646	28.786	25.344
LightGBM	25.674	22.162	20.809	28.735	20.870	27.753	26.991	25.675	24.831	28.489	25.199
LSTM	29.022	32.144	27.432	35.156	27.844	33.137	35.224	32.128	29.368	31.712	31.317
GRU	29.885	27.598	24.730	39.638	28.564	29.902	35.485	30.092	33.955	36.884	31.673
ResNet-LSTM	26.629	24.595	21.051	30.906	21.717	28.465	27.097	25.552	25.328	28.121	25.946
Stacked	24.113	20.738	20.071	27.916	20.173	25.641	24.489	24.208	23.106	27.008	23.746
ResNet-LSTM											

**Table 7** The experimental results in terms of R-Squared

Model	1	2	3	4	5	6	7	8	9	10	Average
GBDT	0.751	0.757	0.725	0.745	0.754	0.744	0.732	0.775	0.775	0.753	0.751
LightGBM	0.765	0.761	0.713	0.755	0.748	0.750	0.739	0.772	0.776	0.759	0.754
LSTM	0.734	0.685	0.694	0.709	0.702	0.733	0.661	0.747	0.737	0.751	0.715
GRU	0.739	0.713	0.696	0.686	0.677	0.742	0.654	0.745	0.712	0.696	0.706
ResNet-LSTM	0.763	0.746	0.735	0.741	0.749	0.766	0.749	0.790	0.786	0.779	0.760
Stacked	0.806	0.808	0.782	0.788	0.803	0.804	0.798	0.817	0.821	0.812	0.804
ResNet-LSTM											

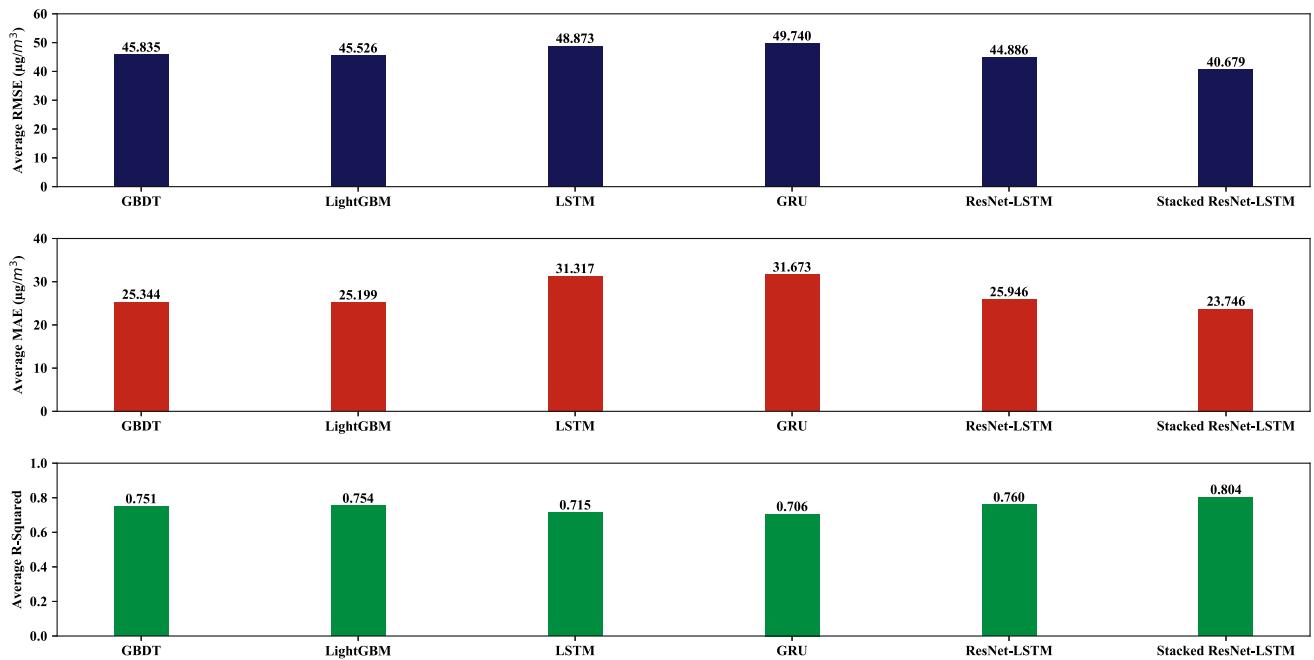
prediction performance than its variant GRU according to the average values of the three metrics. This is probably because LSTM has a better model expression performance than GRU under the abundant training samples in the training dataset.

With the location *Nongzhanguan* (#6) as an example, Fig. 12 illustrates the scatter plots of actual observed PM<sub>2.5</sub> values and corresponding prediction results generated by each model. The areas exhibiting high scatter plot density of LSTM and GRU are slightly concentrated above the 1:1 line, i.e., the predicted PM<sub>2.5</sub> is significantly higher than the real value when the observed PM<sub>2.5</sub> is at a low concentration. Figure 13 presents the forecast result for 1000 h from 2016-12-10 to 2017-01-20 in *Huairou* as a time series, where the black line refers to the observed values, the other colored lines are the prediction results of each model.

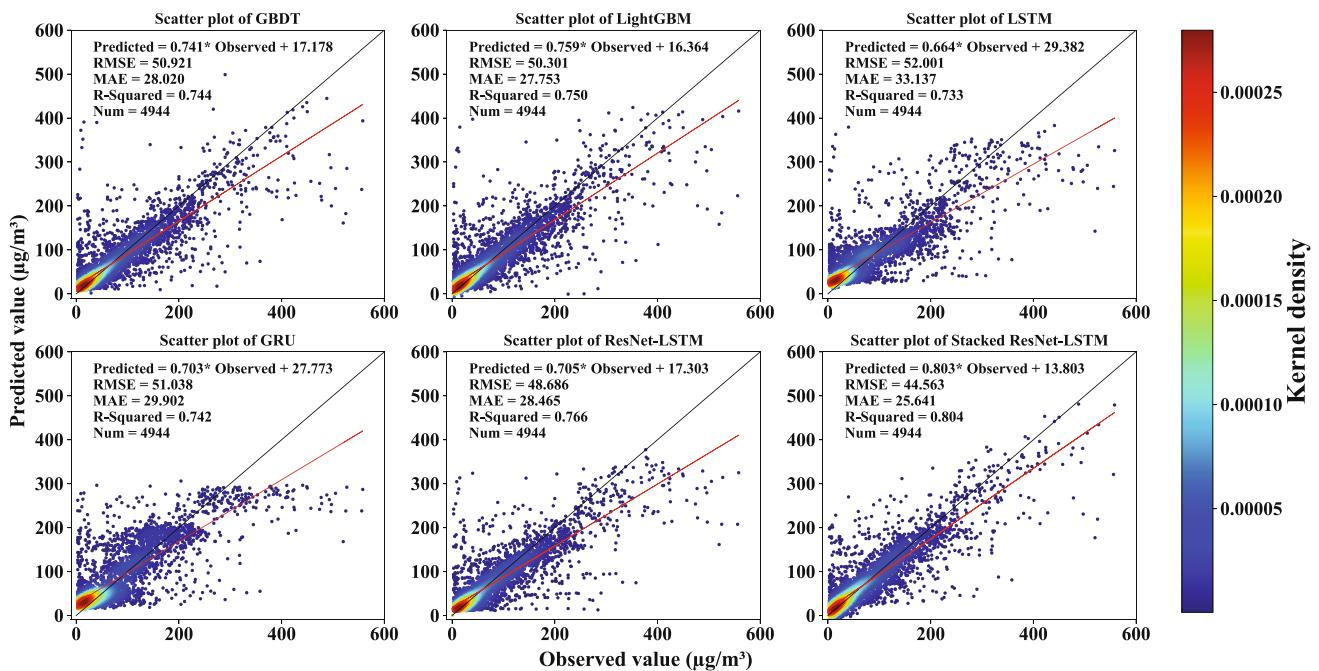
The analysis of the prediction results and comparison to the observations suggests that classical boosting algorithms and neural network models can effectively track the future changing trend of PM<sub>2.5</sub> concentration values. According to Fig. 13, there is a significant deviation in the prediction results of LSTM (Green line) and GRU (Pink line), when the PM<sub>2.5</sub> pollution concentration is relatively low.

To more effectively analyze the forecasting results under different observed values, the air quality is classified according to PM<sub>2.5</sub> concentration levels as given as follows:

1. *Good* PM<sub>2.5</sub> does not exceed 75 µg/m<sup>3</sup>
2. *Mild or moderate pollution* PM<sub>2.5</sub> is between 75 µg/m<sup>3</sup> and 150 µg/m<sup>3</sup>
3. *Severe pollution* PM<sub>2.5</sub> is greater than 150 µg/m<sup>3</sup>



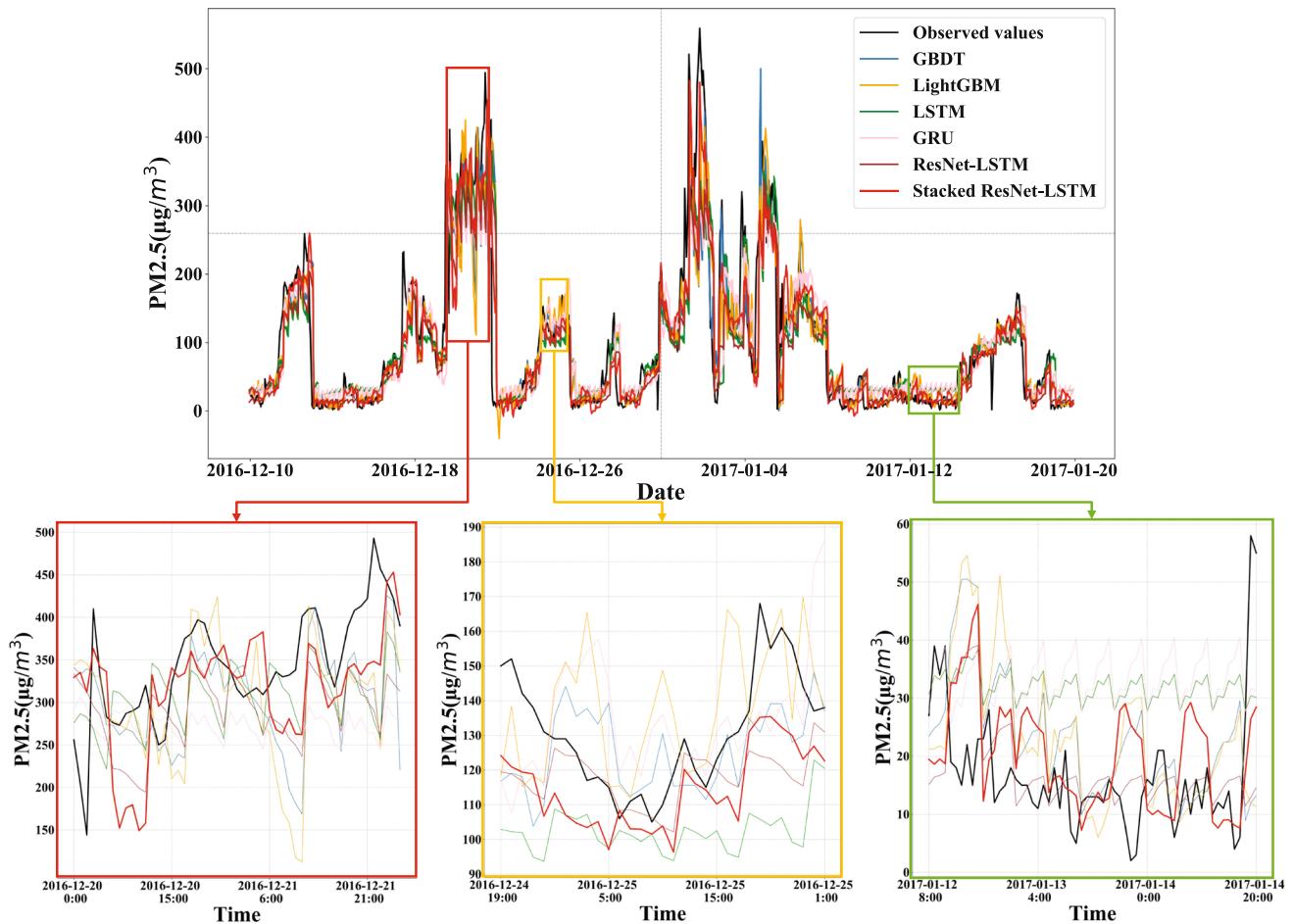
**Fig. 11** The comparison of average RMSE, MAE and R-Squared between Stacked ResNet-LSTM and other comparative models



**Fig. 12** Scatter density graph for observed and predicted  $\text{PM}_{2.5}$  values between Stacked ResNet-LSTM and other comparative models for location *Nongzhanguan*. The redder the color, the denser the scatter

The red, orange, and green boxes in Fig. 13 present the detailed comparison results of the respective model under the mentioned three air quality situations. According to the green box, the prediction results of LSTM and GRU present a cyclical fluctuation state when the observed  $\text{PM}_{2.5}$  values are lower than  $75 \mu\text{g}/\text{m}^3$ . However, this

phenomenon occurs only very slightly in the ResNet-LSTM model. Compared with LSTM and GRU, the prediction of the ResNet-LSTM model fits the observed  $\text{PM}_{2.5}$  value to a higher extent. It is therefore inferred that for recurrent neural networks, feature extraction operation is capable of effectively avoiding model failure under low



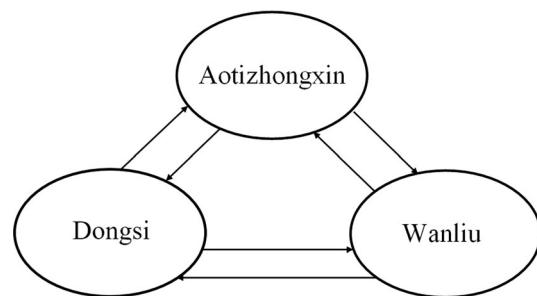
**Fig. 13** Prediction results of different models and their comparison

concentration prediction. Among all the models, the performance of GBDT, LightGBM, ResNet-LSTM and stacked ResNet-LSTM is best when air quality is in good condition. However, it is still noteworthy that most algorithms cannot comply with the real trend when it increases rapidly in a short period. Furthermore, as indicated from this phenomenon, great difficulties remain in accurately predicting PM<sub>2.5</sub> changes. The yellow box presents the comparison results of each model under mild or moderate pollution. It can be seen that the prediction result of the LightGBM model (orange line) has three obvious spikes, which is far from the actual observed values. The other models are capable of predicting the changing trend of PM<sub>2.5</sub> in this concentration range to a better extent. The red box presents the results of each model when the PM<sub>2.5</sub> concentration value exceeds 150 µg/m<sup>3</sup>. As already revealed from the green box, the prediction results of LSTM, GRU and ResNet-LSTM show periodic oscillations, which demonstrates that they cannot achieve accurate forecasting under severe pollution. Nevertheless, after the basic model is integrated, the ensemble model stacked ResNet-LSTM effectively avoids the mentioned

phenomenon and demonstrates better nonlinear fitting ability even at high pollution levels.

### 5.3.2 Comparative experiments of the CORAL model

In this study, three locations, i.e., *Aotizhongxin* (#1), *Dongsi* (#4) and *Wanliu* (#9), are taken as research objects (Fig. 14). In the respective experiment, two of the three above locations are selected as experimental objects to



**Fig. 14** Experimental locations of the transfer learning model. The tail of the arrow represents the source domain. The head of the arrow represents the target domain

compare the prediction results between the transfer learning methods and the ResNet-LSTM basic model. Different from the transfer learning models, ResNet-LSTM model directly used the training dataset from the target prediction area for training.

Table 8 lists the result statistics of all comparative experiments. According to three evaluation metrics, the performance of the ResNet-LSTM model is the optimal generally, followed by CORAL and TCA last. This is not surprising since ResNet-LSTM-as a supervised learning model-is capable of training a model with high prediction accuracy given sufficient training data. In the actual use-case of our underlying assumption (not sufficient training data available in target area), the ResNet-LSTM model could not be trained and therefore would not exist, but in our experiment we do have the training data and therefore can provide the ResNet-LSTM metrics for a comparative analysis with our CORAL and TCA results. In several cases (e.g. *Aotizhongxin* as the source domain and *Wanliu* as the target domain), the R-squared value of the CORAL model prediction result reaches 0.715, close to the ResNet-LSTM model prediction result of 0.763. The forecasting results of *Aotizhongxin* as target domain and *Wanliu* as source domain are taken as an example for illustration in Fig. 15. As revealed from the scatter density graph, the prediction results of CORAL at low concentrations are more concentrated on the 1:1 line in comparison to ResNet-LSTM, although ResNet-LSTM has better performance according to the evaluation indicators. The fitting curves of different models in *Aotizhongxin* are plotted in Fig. 16. According to those, all models are capable of basically tracking changes of PM<sub>2.5</sub> concentration. At some individual time points, the prediction results of the CORAL model and the TCA model show large deviations. But in relation to the training data from the source domain, they already give a relatively accurate forecast performance.

As suggested from the mentioned three sets of comparative experiments, the ResNet-LSTM air quality prediction model is the optimal under sufficient experimental data in the target domain. However, for insufficient experimental data in the target domain, using transfer learning methods is considered to be more effective. In some specific scenarios, the prediction results of the CORAL method by adopting source domain data for training are significantly close to the supervised learning model ResNet-LSTM (e.g., transferring from *Aotizhongxin* to *Wanliu*). For the comparison of the two transfer learning methods, the CORAL method results outperform those achieved by using the TCA method, independent of which source domain and target domain are combined. Lastly, when building a transfer learning model for the target domain, a suitable source domain should be selected, thereby positively impacting the prediction results [42]. Here, we use the CORAL loss as a metric of the distance between the second-order statistics of the source and target features [43].

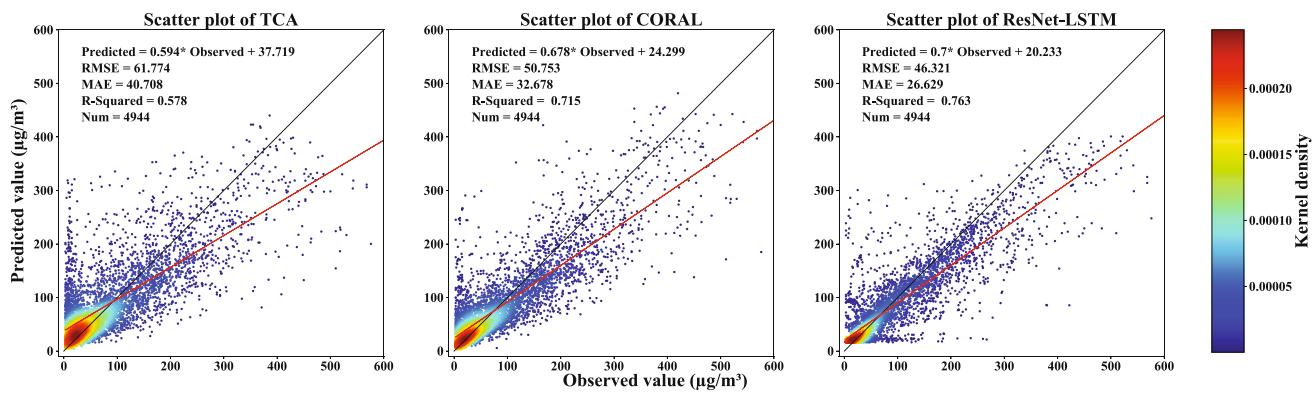
$$\ell_{\text{CORAL}} = \frac{1}{4d^2} \|C_s - C_t\|_F^2, \quad (5)$$

where  $d$  represents the dimension of the feature.

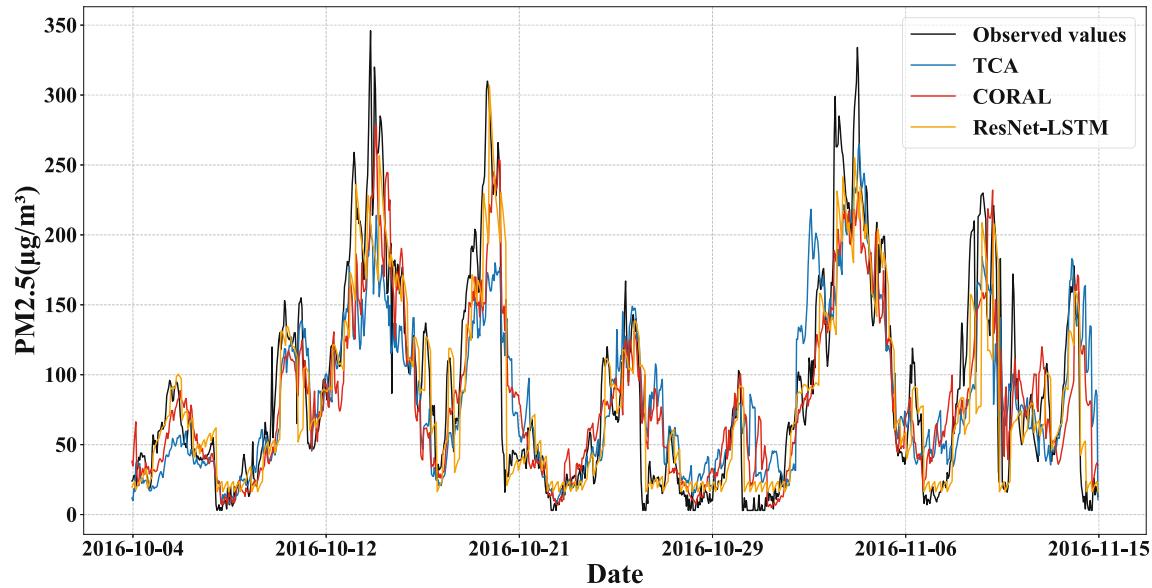
Table 9 presents the CORAL loss between three different domains. It can be seen from Table 9 that the CORAL loss between *Aotizhongxin* and *Wanliu* is the smallest (1098.699), and as is shown in Table 8, the three evaluation metrics between these two domains are also the best in the comparative experiments. Besides, if choosing *Wanliu* as the target domain, *Aotizhongxin*-which has a smaller CORAL loss with *Wanliu* than *Dongsi* with *Wanliu*-performs better in the transfer learning prediction. Taking into account the analyses above, we can come to the conclusion that the smaller CORAL loss, the better prediction performance. When utilizing transfer learning to

**Table 8** Results of comparative experiments for different models compared to the CORAL model at three different locations, *Huairou*, *Aotizhongxin*, and *Changping*

Model	Metrics	Aotizhongxin—Dongsi	Aotizhongxin—Wanliu	Dongsi—Aotizhongxin	Dongsi—Wanliu	Wanliu—Aotizhongxin	Wanliu—Dongsi	Average
TCA	RMSE	69.739	58.123	61.389	61.031	61.774	71.959	64.003
	MAE	46.529	39.242	42.106	42.433	40.708	47.744	43.127
	R <sup>2</sup>	0.547	0.607	0.583	0.567	0.578	0.518	0.566
CORAL	RMSE	61.887	48.051	52.105	51.590	50.753	62.630	54.503
	MAE	38.849	31.299	34.481	34.126	32.678	40.436	35.312
	R <sup>2</sup>	0.643	0.731	0.700	0.690	0.715	0.635	0.686
ResNet-LSTM	RMSE	52.680	42.920	46.321	42.920	46.321	52.680	47.307
	MAE	30.906	25.328	26.629	25.328	26.629	30.906	27.621
	R <sup>2</sup>	0.741	0.786	0.763	0.786	0.763	0.741	0.763



**Fig. 15** Scatter density graph for observed and predicted PM<sub>2.5</sub> values between the CORAL model and other comparative models



**Fig. 16** Prediction results of the CORAL model and comparative models

**Table 9** CORAL loss between three domains

Metrics	Aotizhongxin and Dongsi	Aotizhongxin and Wanliu	Dongsi and Wanliu
CORAL loss	5315.257	1098.699	5550.103

make a prediction on a target area, it is preferred to choose the location with the smallest CORAL loss as the source domain.

## 6 Conclusion

In order to take air pollution forecasting accuracy to the next level, this study constructed an ensemble deep neural network prediction model—the stacked ResNet-LSTM model. In this article, we use ResNet to process the high-

dimensional data to forecast the PM<sub>2.5</sub> concentration 6 h into the future based on historical air quality data and meteorological information. The novel network architecture enables us to explore time-related features and statistical features and provides an accurate prediction on the future PM<sub>2.5</sub> concentration. Moreover, an ensemble method is utilized to increase the prediction accuracy. With PM<sub>2.5</sub> and meteorology data from Beijing, a case study has been performed to verify the superiority of the proposed model. To be specific, the stacked ResNet-LSTM model outperforms the other individual models in all prediction areas of our case study in Beijing, achieving a maximal average R-squared value of 0.804 followed by ResNet-LSTM (0.760), LightGBM (0.754), GBDT (0.751), LSTM (0.715), as well as GRU (0.706). The other evaluation metrics RMSE and MAE of stacked ResNet-LSTM are the lowest among all the comparative models, which indicates a better performance. For the stacked ResNet-LSTM

model, its practicality and generalization for forecasting the PM<sub>2.5</sub> concentration are verified here.

Furthermore, to solve the problem of data deficit for PM<sub>2.5</sub> prediction in areas of interest with insufficient data availability, this study employed a domain adaptation-based model, termed as CORAL. This model is capable of effectively improving PM<sub>2.5</sub> prediction in data shortage scenarios, and its experimental performance is significantly close to the supervised learning model ResNet-LSTM in several scenarios. It is noteworthy that it is also critical to select a suitable location as the source domain of the CORAL model, which can significantly improve the forecasting results of the target predicted area.

Nowadays, deep learning models are also facing privacy and security issues. The most representative existing privacy threats include model extraction attacks and model inversion attacks [44]. A model inversion attack uses APIs (Application Programming Interfaces) provided by the machine learning system to obtain preliminary information of the model and performs reverse analysis on the model through those preliminary information to further obtain private data inside the model or to even reconstruct training data samples [45]. In a model extraction attack, an attacker infers the parameters or functions of the model by sending data in a loop and observing the model results, thereby replicating a machine learning model with similar or even identical functions [46]. In the future, we would like to investigate some privacy-preserving technologies and integrate them into our proposed models to protect the training data and model parameters from hacking.

## Appendix

In this chapter, the process of deriving the linear transformation matrix  $A$  as used in Sect. 4.2.1 is shown in detail.

Conducting the singular value decomposition (SVD) on  $C_s$  and  $C_t$ , thus  $C_s = U_s \Sigma_s U_s^\top$  and  $C_t = U_t \Sigma_t U_t^\top$  can be obtained, where  $U_s$  is a left singular vector,  $U_s^\top$  is a right singular vector, and  $\Sigma_s$  is the diagonal matrix of singular values. The optimal solution to Eq. 1 is as follows:

$$C_s = U_{t[1:r]} \Sigma_{t[1:r]} U_{t[1:r]}^\top, \quad (6)$$

where  $r = \min(r_{C_s}, r_{C_t})$ . Let  $C_s = A^\top C_s A$ :

$$A^\top C_s A = U_{t[1:r]} \Sigma_{t[1:r]} U_{t[1:r]}^\top, \quad (7)$$

where the diagonal elements of  $\Sigma_{t[1:r]}$  are  $r$  largest singular values, the columns of  $U_{t[1:r]}$  are the corresponding left-singular vectors and the rows of  $U_{t[1:r]}^\top$  are the corresponding right-singular vectors. Because  $C_s = U_s \Sigma_s U_s^\top$ , Eq. 7 can be rewritten as:

$$A^\top U_s \Sigma_s U_s^\top A = U_{t[1:r]} \Sigma_{t[1:r]} U_{t[1:r]}^\top. \quad (8)$$

Then, it gives:

$$(U_s^\top A)^\top \Sigma_s (U_s^\top A) = U_{t[1:r]} \Sigma_{t[1:r]} U_{t[1:r]}^\top. \quad (9)$$

Let  $E = \Sigma_s^{+\frac{1}{2}} U_s^\top U_{t[1:r]} \Sigma_{t[1:r]}^{\frac{1}{2}} U_{t[1:r]}^\top$ , where  $E$  is the Moore-Penrose pseudoinverse of  $\Sigma_s^{+\frac{1}{2}}$ . Then the right hand side of Eq. 9 can be written as follows:

$$(U_s^\top A)^\top \Sigma_s (U_s^\top A) = E^\top \Sigma_s E. \quad (10)$$

Lastly, the optimal solution of  $A$  is yielded.

$$A = U_s E = \left( U_s \Sigma_s^{+\frac{1}{2}} U_s^\top \right) \left( U_{t[1:r]} \Sigma_{t[1:r]}^{\frac{1}{2}} U_{t[1:r]}^\top \right). \quad (11)$$

**Acknowledgements** This work was supported by the Professorship of Environmental Sensing and Modeling grant funded by the German Research Foundation (DFG) under Grant Nr. 419317138.

**Funding** Open Access funding enabled and organized by Projekt DEAL.

## Declarations

**Conflict of interest** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Kostka G, Nahm J (2017) Central-local relations: recentralization and environmental governance in china. China Q 231:567–582
2. Curtis L, Rea W, Smith-Willis P, Fenyves E, Pan Y (2006) Adverse health effects of outdoor air pollutants. Environ Int 32(6):815–830
3. Leikauf GD, Kim SH, Jang AS (2020) Mechanisms of ultrafine particle-induced respiratory health effects. Exp Mol Med 52(3):329–337
4. Wang X, Wang B (2019) Research on prediction of environmental aerosol and PM<sub>2.5</sub> based on artificial neural network. Neural Comput Appl 31(12):8217–8227
5. Zhang B, Zhang H, Zhao G, Lian J (2020) Constructing a PM<sub>2.5</sub> concentration prediction model by combining auto-encoder with bi-lstm neural networks. Environ Model Softw 124:104600

6. Tao G, Chen H, Li W (2020) Beijing PM<sub>2.5</sub> influencing factors analysis based on gam. In: 2020 IEEE/WIC/ACM International Joint Conference on web intelligence and intelligent agent technology (WI-IAT), IEEE, p 916–921
7. Iskandaryan D, Ramos F, Trilles S (2020) Air quality prediction in smart cities using machine learning technologies based on sensor data: a review. *Appl Sci* 10(7):2401
8. Dong M, Yang D, Kuang Y, He D, Erdal S, Kenski D (2009) PM<sub>2.5</sub> concentration prediction using hidden semi-markov model-based times series data mining. *Expert Syst Appl* 36(5):9046–9055
9. Dong Y, Wang H, Zhang L, Zhang K (2016) An improved model for PM<sub>2.5</sub> inference based on support vector machine. In: 2016 17th IEEE/ACIS International Conference on software engineering, artificial intelligence, networking and parallel/distributed computing (SNPD), IEEE, p 27–31
10. Fu M, Wang W, Le Z, Khorram MS (2015) Prediction of particular matter concentrations by developed feed-forward neural network with rolling mechanism and gray model. *Neural Comput Appl* 26(8):1789–1797
11. Chen Y (2018) Prediction algorithm of PM<sub>2.5</sub> mass concentration based on adaptive bp neural network. *Computing* 100(8):825–838
12. Xie H, Ji L, Wang Q, Jia Z (2019) Research of PM<sub>2.5</sub> prediction system based on cnns-gru in Wuxi urban area. In: IOP Conference series: earth and environmental science. IOP Publishing, vol 300, p 032073
13. Singh KP, Gupta S, Kumar A, Shukla SP (2012) Linear and nonlinear modeling approaches for urban air quality prediction. *Sci Total Environ* 426:244–255
14. Foresman TW (1998) The history of geographic information systems: perspectives from the pioneers, vol 397. Prentice Hall PTR, Upper Saddle River
15. Campbell JB, Wynne RH (2011) Introduction to remote sensing. Guilford Press
16. Enge PK (1994) The global positioning system: Signals, measurements, and performance. *Int J Wirel Inf Netw* 1(2):83–105
17. Wang J, Christopher SA (2003) Intercomparison between satellite-derived aerosol optical thickness and PM<sub>2.5</sub> mass: Implications for air quality studies. *Geophys Res Lett* 30(21)
18. Han X, Cui X, Ding L, Li Z (2019) Establishment of PM<sub>2.5</sub> prediction model based on MAIAC AOD data of high resolution remote sensing images. *Int J Pattern Recognit Artif Intell* 33(03):1954009
19. Ma Z, Hu X, Huang L, Bi J, Liu Y (2014) Estimating ground-level PM<sub>2.5</sub> in china using satellite remote sensing. *Environ Sci Technol* 48(13):7436–7444
20. Xi X, Wei Z, Xiaoguang R, Yijie W, Xinxin B, Wenjun Y, Jin D (2015) A comprehensive evaluation of air pollution prediction improvement by a machine learning method. In: 2015 IEEE International Conference on service operations and logistics, and informatics (SOLI), IEEE, pp 176–181
21. Saide PE, Carmichael GR, Spak SN, Gallardo L, Osses AE, Mena-Carrasco MA, Pagowski M (2011) Forecasting urban PM<sub>10</sub> and PM<sub>2.5</sub> pollution episodes in very stable nocturnal conditions and complex terrain using wrf-chem co tracer model. *Atmos Environ* 45(16):2769–2780
22. Hong J, Mao F, Min Q, Pan Z, Wang W, Zhang T, Gong W (2020) Improved PM<sub>2.5</sub> predictions of wrf-chem via the integration of Himawari-8 satellite data and ground observations. *Environ Pollut* 263:114451
23. Wang W, Guo Y (2009) Air pollution PM<sub>2.5</sub> data analysis in los angeles long beach with seasonal Arima model. In: 2009 International Conference on energy and environment technology, IEEE, vol 3, pp 7–10
24. Zhang L, Lin J, Qiu R, Hu X, Zhang H, Chen Q, Tan H, Lin D, Wang J (2018) Trend analysis and forecast of PM<sub>2.5</sub> in Fuzhou, china using the Arima model. *Ecol Ind* 95:702–710
25. Deters JK, Zalakeviciute R, González M, Rybarczyk Y (2017) Modeling PM<sub>2.5</sub> urban pollution using machine learning and selected meteorological parameters. *J Electr Comput Eng* 2017:5106045:1–5106045:14
26. Zhu D, Cai C, Yang T, Zhou X (2018) A machine learning approach for air quality prediction: Model regularization and optimization. *Big Data Cogn comput* 2(1):5
27. Li S, Li Y (2013) Nonlinearly activated neural network for solving time-varying complex Sylvester equation. *IEEE Trans Cybern* 44(8):1397–1407
28. Wang X, Yuan J, Wang B (2021) Prediction and analysis of PM<sub>2.5</sub> in fulling district of Chongqing by artificial neural network. *Neural Comput Appl* 33(2):517–524
29. Huang CJ, Kuo PH (2018) A deep cnn-lstm model for particulate matter (PM<sub>2.5</sub>) forecasting in smart cities. *Sensors* 18(7):2220
30. Yeo I, Choi Y, Lops Y, Sayeed A (2021) Efficient PM<sub>2.5</sub> forecasting using geographical correlation based on integrated deep learning algorithms. *Neural Comput Appl* 33(22):15073–15089
31. Choi H, Ryu S, Kim H (2018) Short-term load forecasting based on resnet and lstm. In: 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), IEEE, pp 1–6
32. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, pp 770–778
33. Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780
34. Zeiler MD (2012) Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*
35. Wolpert DH (1992) Stacked generalization. *Neural Netw* 5(2):241–259
36. Sun B, Feng J, Saenko K (2017) Correlation alignment for unsupervised domain adaptation. In: Csurka G (ed) Domain adaptation in computer vision applications. Springer, pp 153–171
37. Friedman JH (2001) Greedy function approximation: a gradient boosting machine. *Ann Stat* 29(5):1189–1232
38. Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu TY (2017) Lightgbm: a highly efficient gradient boosting decision tree. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R (eds) Advances in neural information processing systems, Curran Associates, Inc., pp 3146–3154
39. Zhang W, Wang L, Chen J, Xiao W, Bi X (2021) A novel gas recognition and concentration detection algorithm for artificial olfaction. *IEEE Trans Instrum Meas* 70:1–14
40. Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*
41. Pan SJ, Tsang IW, Kwok JT, Yang Q (2010) Domain adaptation via transfer component analysis. *IEEE Trans Neural Netw* 22(2):199–210
42. Bascol K, Emonet R, Fromont E (2019) Improving domain adaptation by source selection. In: 2019 IEEE International Conference on Image Processing (ICIP), pp 3043–3047
43. Sun B, Saenko K (2016) Deep coral: correlation alignment for deep domain adaptation. In: European Conference on computer vision, Springer, pp 443–450
44. Liu X, Xie L, Wang Y, Zou J, Xiong J, Ying Z, Vasilakos AV (2020) Privacy and security issues in deep learning: a survey. *IEEE Access* 9:4566–4593

45. Fredrikson M, Jha S, Ristenpart T (2015) Model inversion attacks that exploit confidence information and basic countermeasures. In: Proceedings of the 22nd ACM SIGSAC Conference on computer and communications security, pp 1322–1333
46. Tramèr F, Zhang F, Juels A, Reiter MK, Ristenpart T (2016) Stealing machine learning models via prediction apis. In: 25th

USENIX Security Symposium (USENIX Security 16), pp 601–618

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.