scientific reports

Check for updates

OPEN

# Air-quality prediction based on the ARIMA-CNN-LSTM combination model optimized by dung beetle optimizer

Jiahui Duan, Yaping Gong✉, Jun Luo & Zhiyao Zhao

**Air pollution is a serious problem that affects economic development and people's health, so an efficient and accurate air quality prediction model would help to manage the air pollution problem. In this paper, we build a combined model to accurately predict the AQI based on real AQI data from four cities. First, we use an ARIMA model to fit the linear part of the data and a CNN-LSTM model to fit the non-linear part of the data to avoid the problem of blinding in the CNN-LSTM hyperparameter setting. Then, to avoid the blinding dilemma in the CNN-LSTM hyperparameter setting, we use the Dung Beetle Optimizer algorithm to find the hyperparameters of the CNN-LSTM model, determine the optimal hyperparameters, and check the accuracy of the model. Finally, we compare the proposed model with nine other widely used models. The experimental results show that the model proposed in this paper outperforms the comparison models in terms of root mean square error (RMSE), mean absolute error (MAE) and coefficient of determination ($R^2$). The RMSE values for the four cities were 7.594, 14.94, 7.841 and 5.496; the MAE values were 5.285, 10.839, 5.12 and 3.77; and the $R^2$ values were 0.989, 0.962, 0.953 and 0.953 respectively.**

Due to industrialization, urbanization, and other factors, air pollution has become increasingly prominent. The air quality index (AQI) is an important index reflecting the level of atmospheric pollution[1], and its size is closely related to the content of various pollutants in the atmosphere. There are six major pollutants affecting air quality: PM2.5, $PM_{10}$, $NO_2$, $SO_2$, CO, and $O_3$. Continuous exposure to air pollution can cause a variety of diseases, such as respiratory, cardiovascular, neurological, etc., and its harm is increasing[2,3]. Air pollution is the product of multiple factors, and its concentration has non-constant and nonlinear characteristics, which brings difficulties to the forecast of atmospheric environmental quality indicators.

Many academics have suggested many prediction models in recent years. The statistics model, machine learning model, and deep learning model can all be generically categorized as these three types of models. A statistical model bases its explanation of cause and effect on assumptions about the distribution of the data and places a high emphasis on parameter inference.

The application of statistical methods in air quality prediction mainly includes the autoregressive (AR) model, the autoregressive integrated moving average (ARIMA) model, the gray model, and the multiple linear regression (MLR) model[4,5] proposed an algorithm to assess the pollution level of air quality parameters and create a new air quality index based on the fuzzy reasoning system to predict air quality parameters by AR model. Zhang et al.[6] examines two different approaches to model development, including GAM and traditional linear regression methods. To show the requirement for first-order differencing[7], proposed an ARIMA model based on the augmented Dickey-Fuller test for PM2.5 annual data. In order to offer accurate forecasts that can accurately capture seasonal and nonlinear properties[8], created a seasonally nonlinear gray model to account for seasonal fluctuations in the time series of seasonally fluctuating pollution indicators.

Machine learning models rely on large data sets to predict the future, weakening the convergence problem and focusing on model prediction. Mehmood et al.[9] discusses the transformation of traditional methods into machine learning methods and analyzes emerging trends to identify potentially valuable research directions. Using machine learning models and methods. Varghese and Kumar[10] developed a machine learning-based empirical model to predict the laminar combustion rate of air pollutants under high pressure and high temperature conditions, using volume fraction as the independent variable. Zhang et al.[11] uses machine learning models and methods in order to predict how the unpredictability and variability of the indoor mode can cause

School of Marine Engineer Equipment, Zhejiang Ocean University, Zhoushan, China. ✉email: ypgong@zjou.edu.cn

excessive adjustment or a deficiency of air quality. Rakholia et al.[12] developed a model that included factors such as weather conditions, urban traffic, air quality data in residential and industrial areas, urban spatial information, time-series composition, and pollution concentrations. Gu et al.[13] proposes a new hybrid interpretable prediction machine learning model for PM2.5 prediction that can outperform other models in terms of peak prediction accuracy and model interpretability. Maltare and Vahora[14] focuses on support vector machine algorithms based on RBF kernel models. Munir[15] used machine learning models to assess the impact of intelligent transport interventions on air quality.

Deep learning model is more adaptive and easily transformable than a machine learning model, allowing easier adaptation to different domains and applications. Zhang et al.[16] made a comprehensive review of the Deep Learning Method dedicated to air pollution concentration forecasting. Wu et al.[17] proposes a hybrid deep learning-based model to predict the pollutant concentration in the next hour at the network scale based on the identified spatio-temporal features. Jurado[18] developed a fast and accurate system using convolutional networks for real-time forecasting of air pollution based on wind speed, traffic flow, and building geometry. Zhang et al.[19] gives a meta-learning algorithm for knowledge transfer between cities with large differences that can incorporate spatio-temporal correlations between monitoring stations and transfer data from other cities with rich training data. Saez and Barceló[20] proposed a new model that can predict the space for long-term and short-term exposure to air pollutants and has relatively low monitoring STA pollutants and a lower calculation time.

Some scholars have developed combinatorial models to improve the accuracy of prediction in pursuit of greater prediction rates. Due to the combination of the advantages of different models, the combined model's accuracy has greatly improved in prediction accuracy. Kshirsagar[21] explores the role that neural networks, regression, and hybrid models play in the analysis, prediction, and mitigation of air pollution, taking into account the most recent developments and new research in the field. Zhang and Li[22] combined the efficient features of CNN with the algorithmic advantages of LSTM to propose a CNN-LSTM model to predict future air pollution data. Vlachokostas[23] confirmed the regression model by using multiple stepwise regression analysis to find a significant statistical relationship between $C_6H_6$ and CO. Gunasekar et al.[24] developed a new hybrid model for air quality prediction, optimizing the residual error of ARIMA by the LSTM algorithm. Wang et al.[25] added an attention mechanism to the model to improve the prediction accuracy of the LSTM model. Dai et al.[26] established five haze hazard risk assessment models by improving theparticle swarm optimization (IPSO) light gradient boosting machine (LightGBM) algorithm and a hybrid model combining XGBoost, four GARCH models and MLP model (XGBoost-GARCH-MLP) is proposed to predict PM2.5 concentration values and volatility[27].

With the rapid development of soft computing technologies, many meta-heuristic algorithms have recently been designed and used as competitive alternative solutions to address improved accuracy of predictive models due to their simplicity and ease of implementation.Grey Wolf Optimizer (GWO) is a nature-inspired optimisation algorithm inspired by the behaviour of grey wolves in packs. Its flexibility and efficiency make it a popular optimisation algorithm. Akilandeswari et al.[28] used LSTM with the Weighted Grey Wolf Optimizer (LSTM-WGWO) to increase the accuracy of the air quality index significantly.The Harris-hawks optimisation algorithm is a nature-inspired group intelligence based optimisation algorithm where the objective is to minimise or maximise an objective function given a constraint. Du et al.[29] proposes a new multi-objective optimisation version of HHO and develops a new hybrid model to improve the accuracy of the predictive model. PSO is a population intelligence based optimisation algorithm inspired by the behaviour of groups of organisms searching for optimal solutions in the solution space. Huang et al.[30] improved the PSO algorithm accordingly, optimized the overall prediction performance of BP neural network, adjusted the change strategy of the inertia weight as well as the learning factor, and ensured the diversity of particles during the early stage and the fast convergence to the global optimal solution.The Cuckoo optimisation algorithm is an optimisation algorithm based on the idea of parasitism in a bird's nest, simulating the biology of a male bird occupying a nest and a hetero bird making the same random behaviour and thus searching for the optimal solution. Sun et al[31] proposes a hybrid model for cuckoo search optimisation based on principal component analysis (PCA) and least squares support vector machine (LSSVM). The model outperforms a single LSSVM model with default parameters and a general regression neural network (GRNN) model for PM2.5 concentration prediction.In summary, machine learning and deep learning models can handle time series forecasting more accurately than traditional statistical models. However, due to the non-stationary nature of AQI data, it may be difficult for individual models to fully explore the internal regularities among the data. Most of the comparative models chosen by previous models are based on derivatives of the proposed model, do not provide a comprehensive comparison of other models, and have limited accuracy. A new combined model is therefore proposed in this paper. To verify the superiority of the model, the AQI is used as an example for forecasting and four different cities in China are selected for the study to compare the forecasting effectiveness of other models. Beijing, Lanzhou, Jiaozuo and Guangzhou were chosen for the study.

The main contributions of this paper are as follows: (1) The linear part of the data is extracted and fitted using the ARIMA model to output the prediction results of the linear part and the non-linear part, and the output non-linear part is imported into the deep learning model for fitting to obtain the prediction values of the non-linear part. (2) The prediction results of the linear part and the non-linear part are combined to obtain the final prediction output. (3) To avoid the problem of blindness in CNN-LSTM hyperparameter setting, this paper uses a dung beetle optimization algorithm to search for hyperparameters of the CNN-LSTM model.

## Materials and methods

**Statistical method.** The ARIMA model is called the average model of the Returning Integration Movement, which is usually written as ARIMA(p, d, q). This model is able to handle non-stationary series and is widely used in algorithmic prediction and has high accuracy in air quality prediction. In the ARIMA model, AR is the autoregressive and p is the number of autoregressive terms; I is the difference and d is the number of

differences (order) made to make it a smooth series; MA is the sliding average and q is the number of sliding average terms, and the mathematical model can be represented by (1). In general, second-order differences are single-integer smooth data, i.e., only the ARMA(p, 2, q) model is required. ARMA(p, 2, q) can be transformed into AR() and MA(), which correspond to the characteristic that both functions exhibit a gradual decay. The p's and q's are determined by either the deficit pool information criterion (AIC) or the Bayesian information criterion (BIC). In this paper, BIC is used to determine the values of p and q. The BIC formula is (2).

$$y_t = c + \phi_1 * y_{t-1} + \cdots + \phi_p * y_{t-p} + \theta_1 * e_{t-1} + \cdots \theta_q * e_{t-q} \tag{1}$$

$$B_{BIC} = k * \ln(n) - 2\ln(L) \tag{2}$$

where $y_t$ is the number of difference levels, $c$ is a constant value, $\phi$ is the AR parameter (autocorrelation size), $p$ is the number of lags (AR order), $\theta$ is the MA parameter value (error autocorrelation), $q$ denotes the number of lags (order of the model MA), and $e_t$ is the error[32]. $k$ is the number of model parameters, $n$ is the number of samples, and $L$ is the likelihood function.

**Machine learning model.** *RF.* Random forest is an integrated prediction model based on decision trees, which integrates multiple decision trees, each of which has a certain dependence on the independently sampled random vector values, and all decision trees in the random forest have the same distribution. The two most important parameters of RF are Number of trees and Number of features. Number of decision trees indicates the number of trees in the forest andNumber of features indicates the number of randomly selected features for each decision tree[33]).

*SVM.* A supervised learning algorithm, Support Vector Machine (SVM), is a generalized linear classifier that performs binary classification of data in a supervised learning manner, with its decision boundary being the maximum margin hyperplane of the learned example solution. For example, $\omega \cdot x + b = 0$ is separation hyperplane in Fig. 1.

The SVM model parameters include: kernel function, penalty coefficient, regularization parameter and accuracy. There are five kernel functions: linear, poly, rbf, sigmoid and pre-computed. This paper choose linear kernel function, mathematical formula for (3); maximum number of iterations Number of iterations of the algorithm.

$$K(x, z) = x \cdot z \tag{3}$$

**Deep learning model.** *LSTM.* Long-short-term memory (LSTM), as a unique class of recurrent neural networks, is used to solve the gradient diffusion problem in recurrent neural networks. LSTM model contains three main gates, namely: forget gate, memory gate and output gate, and its structure is shown in Fig. 2.

The task of the forget gate is to accept a long-term memory $C_{t-1}$ (the output from the previous unit module) and decide which part of $C_{t-1}$ to retain and forget, with the mathematical expression (4);

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

The memory gate will forget the attribute information discarded by the gate, locate the corresponding new attribute information in the unit module, and supplement the discarded attribute information. The memory

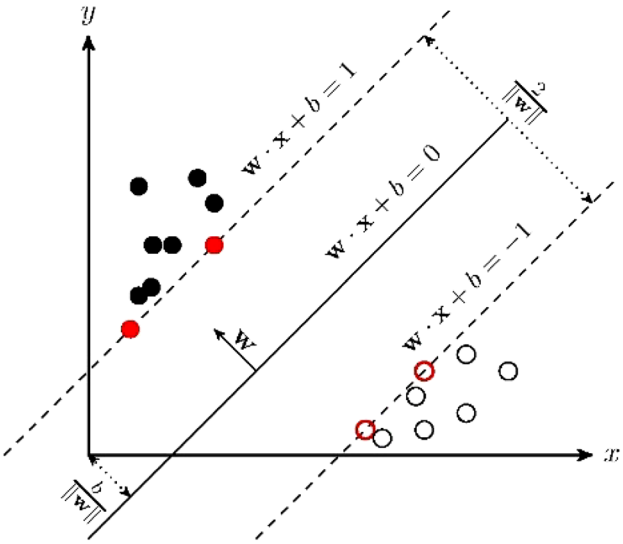



**Figure 1.** SVM schematic.

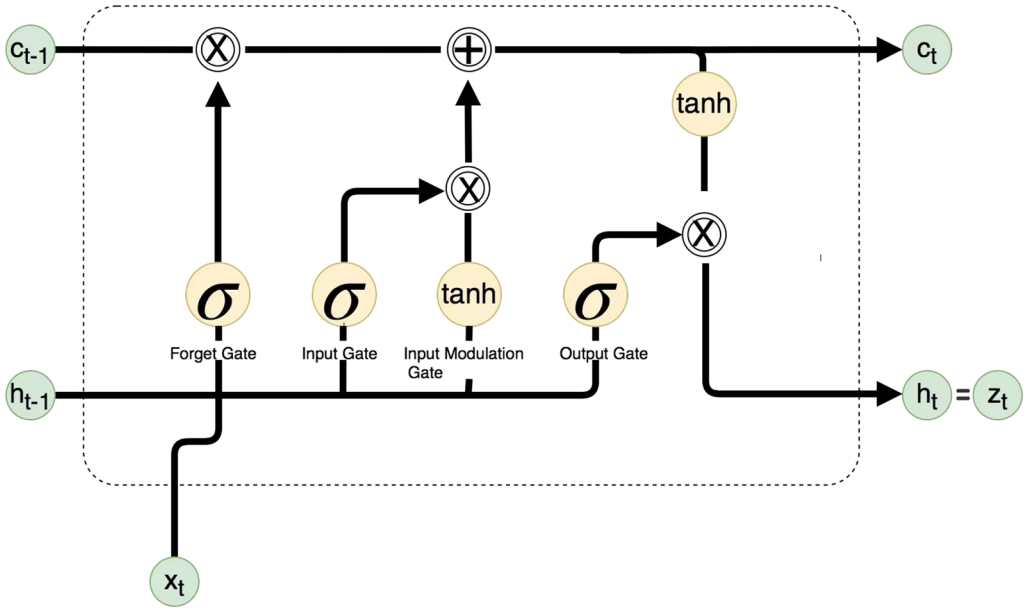



**Figure 2.** LSTM Structure.

gate is made up of two layers: the sigmoid layer and the tanh layer, which have the mathematical expressions (5), (6), and (7).

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

$$\tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \tag{6}$$

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \tag{7}$$

The output gate is used to determine the cell state output part, and the cell state is processed through the tanh layer, and the two are multiplied to get the final information we want to output, with the mathematical expressions (8) and (9).

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{8}$$

$$h_t = o_t \tanh(C_t) \tag{9}$$

where $h_{(t-1)}$ represents the previous cell output, $x_t$ represents the current cell input, $\sigma$ denotes the sigmoid activation function, $W_f$ represents the forgetting gate's weight coefficient matrix, and $b_f$ represents the forget gate bias vector. $W_i$, $b_i$ denote the input gate weight coefficient matrix and bias vector determined by the sigmoid activation function, while $W_c$ and $b_c$ denote the input gate weight coefficient matrix and bias vector determined by the hyperbolic tangent activation function, respectively. *tanh* denotes the hyperbolic tangent activation function. $W_o$, $b_o$, denote the output gate weight coefficient matrix and bias vector, respectively, and $o_t$ denotes the output gate at time $t$[34].

*DBO.* DBO (Dung Beetle Optimizer, DBO for short) is a new population intelligence algorithm based on beetle ball rolling, dancing, foraging, stealing, reproduction and other behaviors. This algorithm is characterized by strong merit-seeking ability and fast convergence. The DBO algorithm consists of four main processes: ball rolling, breeding, foraging and stealing. In the case of dung beetle unobstructed ball rolling, assuming that light intensity affects dung beetle position, the formula for updating the dung beetle's position as follows.$x_i(t+1) = x_i(t) + \alpha \cdot k \cdot x_i(t-1) + b \cdot \Delta x$

$$\Delta x = |x_i(t) - X^w| \tag{10}$$

where $t$ is the current number of iterations, $x_i(t)$ is the position information of the $i$ th praying mantis in the $i$ th iteration, and $k \in (0, 0.2]$ is the current number of iterations represents the deflection coefficient's constant value, $b$ represents the value of the constant assigned to (0, 1), and $\alpha$ represents the natural coefficient assigned to −1 or 1. $X^w$ represents the ball's worst position, and $\Delta x$ is used to simulate the change in light intensity[35].

When the dung beetle encounters an obstacle that prevents it from progressing, it adjusts by dancing to find a new path. The algorithm uses a tangent function to model the dancing behavior. The dung beetle's position is updated as follows after determining a new direction and continuing to roll the ball.

$$x_i(t+1) = x_i(t) + tan(\theta)|x_i(t) - x_i(t-1)| \tag{11}$$

where $\theta$ is an angle tilted from the $[0, \pi]$ direction[35].

In the reproductive process, the scarab algorithm adopts an edge selection strategy to simulate the spawning area of scarabs as Equation.

$$\begin{cases} Lb^* = \max(X^* \cdot (1-R), Lb) \\ Ub^* = min(X^* \cdot (1-R), Ub) \end{cases} \tag{12}$$

where $X^*$ represents the current optimal solution, while $Lb^*$ represents the optimal solution of the optimal solution, and $Ub^*$ represents the optimal solution of the optimal solution. $R = 1 - \frac{t}{T}$ and $T$ is the maximum number of iterations, $Lb$ is the upper and lower limits of the optimal solution and $Ub$ is the upper limit of the optimal solution[35].

When the egg-laying zone is determined, the dung beetle lays only one egg per iteration. It is clear from (12) that the egg-laying area is dynamically adjusted in the iteration and therefore the location of l eggs is also dynamic as Equation[35].

$$B_i(t-1) = X^* + b_1 \cdot (B_i(t) - Lb^*) + b_2 \cdot (B_i(t) - Ub^*) \tag{13}$$

where, $B_i(t)$ is the position of the $i$ th sphere at the $t$ th iteration, $b_1$ and $b_2$ are two independent random vectors of size $1 \times D$, and $D$ is the dimension of the optimal solution[35].

During the predation process, the boundary of the optimal predation area is determined according to the position changes of the insects during the predation process.

$$\begin{cases} Lb^b = \max\left(X^b \cdot (1-R), Lb\right) \\ Ub^b = \min\left(X^b \cdot (1+R), Ub\right) \end{cases} \tag{14}$$

where $X^b$ is the global optimization, $Lb^b$ is the lower bound of the optimal search domain, and $Ub^b$ is the upper bound of the optimal search domain. The location of the little beetle is updated as follows.

$$x_i(t+1) = x_i(t) + C_1 \cdot \left(x_i(t) - Lb^b\right) + C_2 \cdot \left(x_i(t) - Ub^b\right) \tag{15}$$

where, $x_i(t)$ denotes the position information of the $i$th dung beetle at the $t$th iteration, $C_1$ denotes a random number obeying normal distribution, and $C_2$ denotes a random vector belonging to (0, 1)[35].

During the stealing phase, the location of the thieving dung beetle is updated as follows.

$$x_i(t-1) = X^b + S \cdot g \cdot \left(\left|x_i(t) - X^*\right| + \left|x_i(t) - X^b\right|\right) \tag{16}$$

where $x_i(t)$ represents the location information of the $i$th thief at the $t$th iteration, $g$ represents a $1 \times D$ random vector that obeys a normal distribution, and $S$ represents a constant value[35].

**Combination model.** *CEEMDAN-CNN-LSTM and CEEMDAN-LSTM.* CEEMDAN is an improved algorithm based on EMD. CEEMDAN improves the reconstructed signal by adding a limited amount of white noise consistent with the standard normal distribution in each iteration[36]. The CEEMDAN algorithm solves the EMD sub-modal mixing problem and the EEMD and CEEMD residual white noise problem. The algorithm steps are divided into 4 main steps.

The first step is to introduce Gaussian white noise into the known signal $y(t)$ to obtain a new signal $y(t) + (-1)^q \varepsilon \upsilon^j(t)$, where q = 1 or 2. EMD decomposition is performed on the new signal to generate a characteristic mode component $C_1$ in the form of Eq. (17). As shown in, the ensemble average of the N mode components generated in the second step yields the first characteristic mode component of the CEEMDAN decomposition (18). The third step, by Eq. (19), calculates the residual after removing the 1st mode component. The preceding process is repeated in the fourth step until the obtained residual signal is a monotonous function. In this way, the number of eigenmodes can be obtained. The original signal y(t) is then decomposed into (20).

$$E\left(y(t) + (-1)^q \varepsilon \upsilon^j(t)\right) = C_1^j(t) + r^j \tag{17}$$

$$\overline{C_1(t)} = \frac{1}{N}\sum_{j=1}^{N} C_1^j(t) \tag{18}$$

$$r_1(t) = y(t) - \overline{C_1(t)} \tag{19}$$

$$y(t) = \sum_{k=1}^{k} \overline{C_k(t)} + r_k(t) \tag{20}$$

where $E_i(\cdot)$ is the $i$ th eigenmode obtained by EMD decomposition, $\overline{C_i(t)}$ is the $i$th eigenmode obtained by CEEMDAN decomposition, $\upsilon^j$ is composed of Gaussian noise (Gaussian noise) , $j$ is the amount of white noise added, $\varepsilon$ is the white noise standard table, $y(t)$ is the decomposed signal.

CEEMDAN-CNN-LSTM and CEEMDAN-LSTM i.e., the CNN-LSTM model is used to fit each IMF component with the LSTM model, and the final prediction is obtained after combining all components, and the process is shown in Fig. 3.

*ARIMA-CNN-LSTM*、*ARIMA-DBO-LSTM and ARIMA-DBO-CNN-LSTM.* Consider the time series data $x_t$ as the combination of linear component $L_t$ and nonlinear component $N_t$ represented by (21). Since linear and nonlinear modeling methods have their own characteristics, the former can only identify linear features of time series, while the latter can effectively mine them[37]. The ARIMA model can predict short-period linear trends well, while the LSTM model can predict complex, non-linear time series well[32].

The ARIMA model is used to predict the linear and nonlinear components of the data, which is then fed into the deep neural network and fit to obtain the predicted value of the nonlinear component. On this basis, the data of both linear and nonlinear aspects are integrated, and the final prediction result is obtained. In order to overcome the blindness of hyperparameter setting, the dung beetle optimization algorithm is introduced to determine the optimal value of hyperparameter setting, the model flow is shown in Fig. 4.

$$x_t = L_t + N_t \tag{21}$$

## Air pollutant concentration prediction

**Study area selection.** To verify the prediction effect of the model, AQI data of Beijing, Lanzhou, Jiaozuo and Guangzhou cities in China are selected for the study. Because all four cities are industrial, a large number of industrial emissions cause severe air pollution. The Chinese government has worked to reduce urban air pollution in recent years, but China's air quality ranking remains at the bottom. Therefore, the air forecasts for these three cities are very important.

The AQI data for the four cities in this paper were gained by the Resource and Environment Science and Data Center of the Chinese Academy of Sciences (https://www.resdc.cn/Default.aspx), and the data are daily AQI data from January 2015 to March 1, 2022, as shown in Fig. 5.

**Data processing.** On this basis, the training samples are divided into two parts, one part accounts for 80% of the training samples, and the other part accounts for 20% of the test samples. Also, in order to improve the training speed of the model, the data are mapped between (0, 1] by a normalization operation. with the normalization formula as follows.

$$x_i = \frac{x - x_{min}}{xmin_{max}} \tag{22}$$

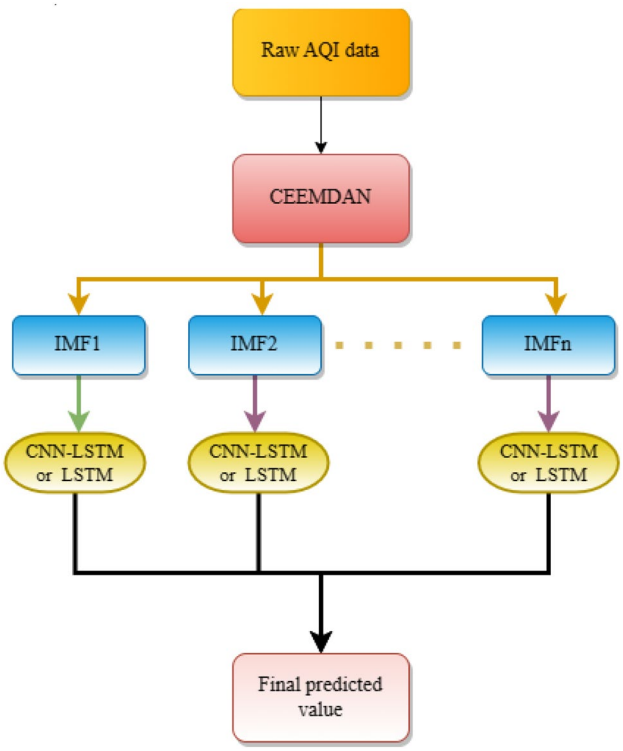



**Figure 3.** Data decomposition model process.

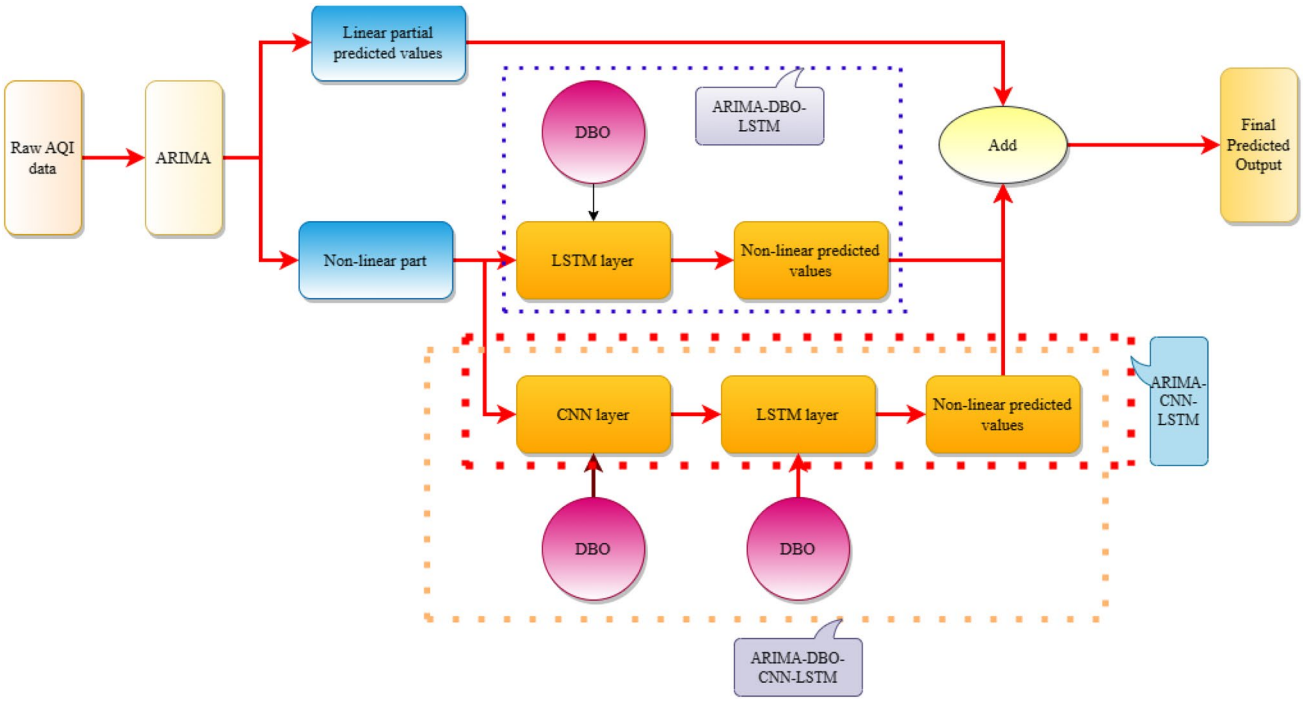



**Figure 4.** ARIMA-DBO-CNN-LSTM model and derived model process.

After model training and prediction, the data need to be subjected to an inverse normalization operation to facilitate the calculation of the evaluation function and plotting, with the inverse normalization equation being as follows.

$$x = (xmin_{max}x_i + x_{min}) \tag{23}$$

Among them, $x_i$ represents the standardized data, $x_{max}$ represents the largest data in the array, and $x_{min}$ represents the smallest data in the array. This paper chooses three evaluation indicators: root mean square error RMSE, coefficient of determination $R^2$, and mean absolute error MAE, and provides specific calculation formulas to accurately compare the prediction effects of each model.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\widehat{y}_i - y_i\right)^2} \tag{24}$$

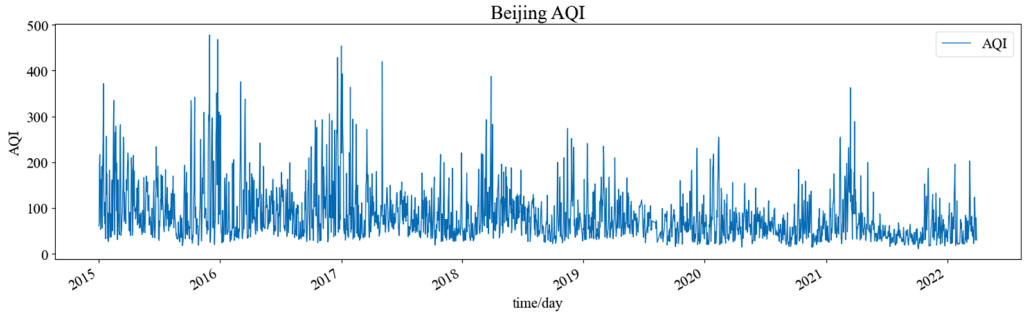
$$R^2 = \frac{\sum_{i=1}^{n}\left(y_i - \widehat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \widehat{y}\right)^2} \tag{25}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\widehat{y}_i - y_i\right| \tag{26}$$

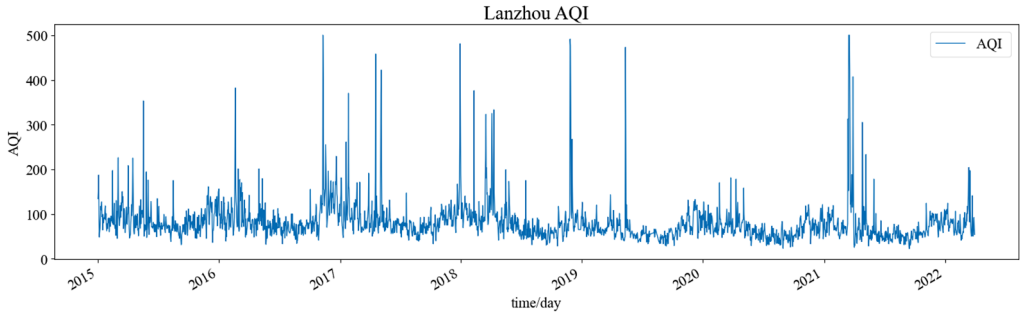
where $n$ is the sample capacity, $y_i$ is the sample value, $\overline{y}$ is the mean value, and $\widehat{y}_i$ is the predicted value.

**model prediction results.** *Model parameter setting.* For the advantages of the ARIMA-DBO-CNN-LSTM model, we compared the classic machine learning model, deep learning model and statistical model, respectively, and the selection of the combined model is not limited to the derivatives of the model proposed in this paper, but also selects the data decomposition combined model which is very widely used at present. And one-dimensional regression equation is used, and multiple experiments are carried out on each equation to ensure that it has the best forecasting effect. On this basis, the maximum constraints on p and q are made using the BIC criterion, and the statistical model restricted the maximum value of p and q to 5. Table 1 shows all parameters in the four cities. The hyperparameter settings of other models are shown in Table 2.

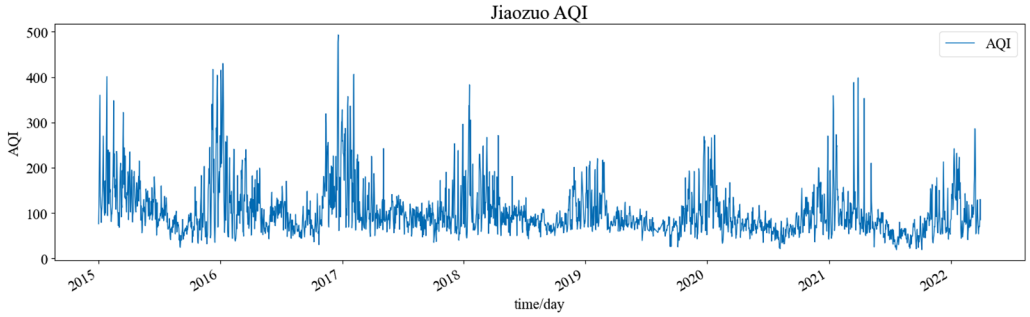
*Forecast comparison and analysis.* All models were predicted after the parameters were set, and the CEEMDAN-CNN-LSTM and CEEMDAN-LSTM models AQI data were decomposed by CEEMDAN to obtain 9 or 10 IMF components and one residual component, as shown in Fig. 6.



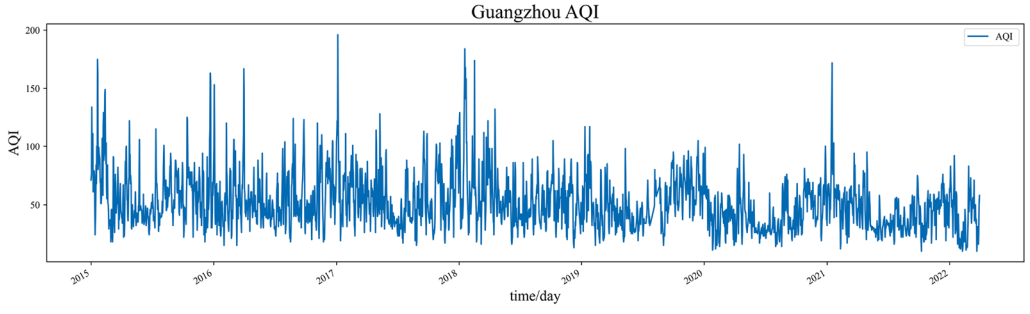

**(a)** Beijing




**(b)** Lanzhou




**(c)** Jiaozuo




**(d)** Guangzhou

**Figure 5.** Raw data.

| City | Beijing | Lanzhou | Jiaozuo | Guangzhou |
|---|---|---|---|---|
| ARIMA parameter setting | (3, 1, 1) | (2, 1, 2) | (3, 1, 1) | (2, 1, 2) |

**Table 1.** Statistical model parameter setting.

| Model type | Models | Parameter setting |
|---|---|---|
| Traditional machine learning models | SVM | kernel = 'linear', Other parameters select default |
| Deep learning models | LSTM | neurons1 = 50, neurons2 = 100, neurons3 = 150, batch_size = 64, epochs = 100, Learning Rate = 0.1, Sliding Window = 10 |
| Combination model | ARIMA-CNN-LSTM | filters = 512, kernel_size = 2, strides = 1, 3 layers of neurons = 50, batch_size = 64, epochs = 100, Learning Rate = 0.2, Sliding Window = 10 |
| | CEEMDAN-CNN-LSTM | filters = 512, kernel_size = 2, strides = 1, neurons = 128, batch_size = 100, epochs = 100, Learning Rate = 0.2, Sliding Window = 10 |
| | CEEMDAN-LSTM | neurons1 = 128, neurons2 = 100, epochs = 100, Learning Rate = 0.2, Sliding Window = 10 |
| Optimization algorithm model | | 3 layers of neurons = [1,300], Sliding Window = [1,50], Learning Rate = [0.001,0.99], batch_size = [1,300], filters = [1,600], kernel_size = [1,10], strides = [1,5], |

**Table 2.** Model parameter settings.

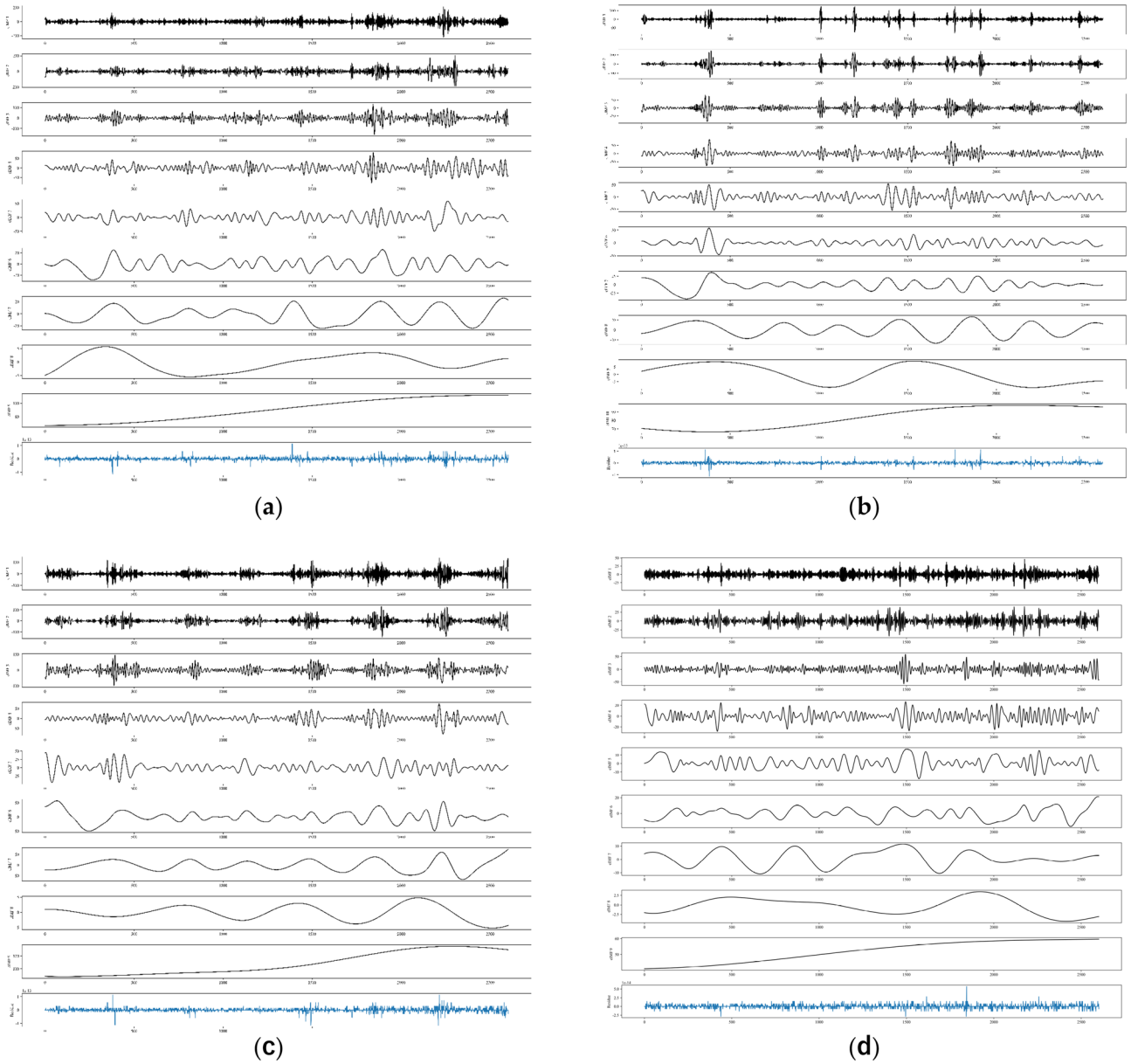
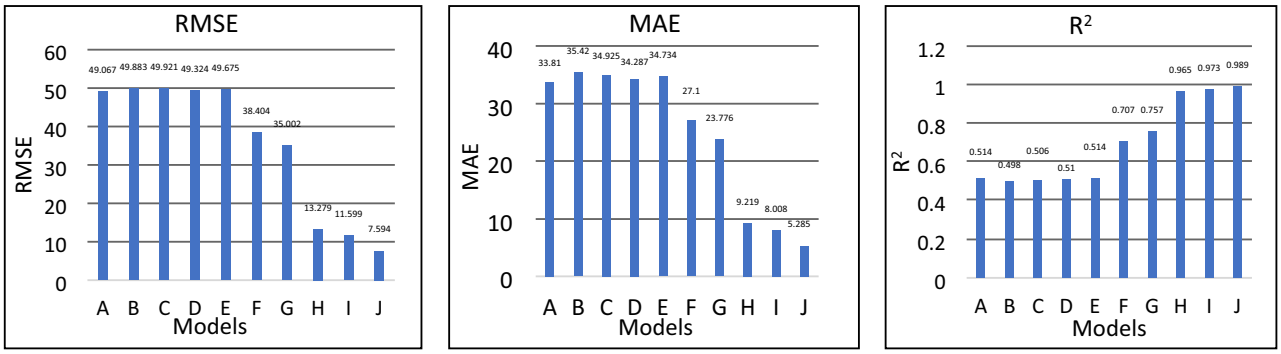


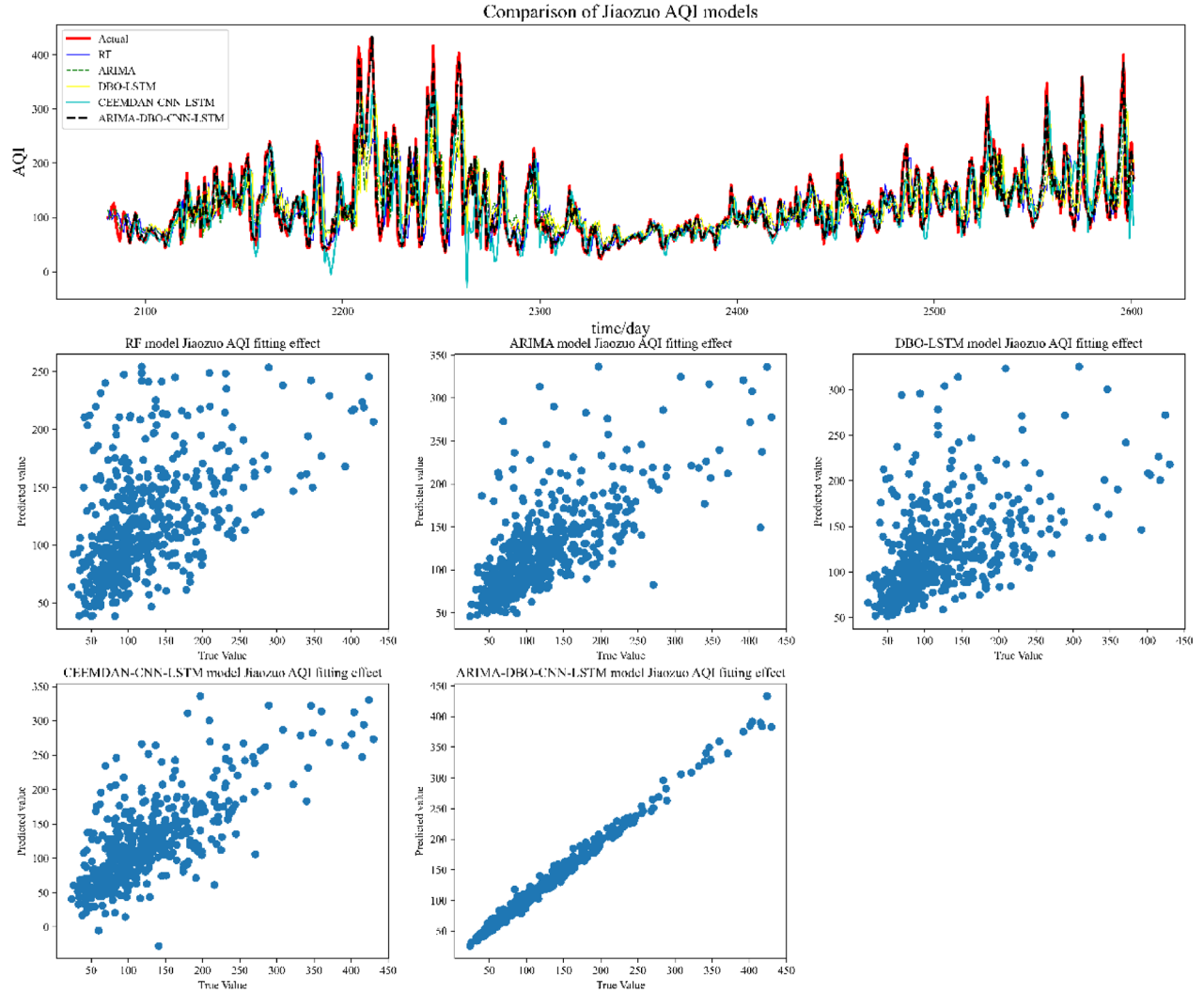

**Figure 6.** CEEMDAN decomposition.

## Results and analysis

Taking Jiaozuo City as an example, the evaluation indicators obtained after import-ing the evaluation function were compared with the output results of all the models after they had been run. In order to compare the performance of the evaluation metrics of dif-ferent models more intuitively, the evaluation metrics of each model were plotted as bar charts as shown in Fig. 7a. In order to clearly compare the ARIMA-DBO-CNN-LSTM model with other kinds of models, the best-performing model among the models was selected to plot line and scatter plots, as shown in Fig. 7




**(a)** Comparison of assessment indicators in Jiaozuo City




**(b)** Jiaozuo City Model Scatter Comparison

**Figure 7.** Jiaozuo City Forecast Comparison.

Through the predicted effect of Jiaozuo City we can easily find:

A. The single model has poor forecasting ability, while the combined model has significantly better forecasting accuracy than the single model.
B. The four types of single prediction models performed similarly, with the $R^2$ metric showing a prediction accuracy of around 0.5, with SVM being the worst performer among the traditional machine learning models.
C. By splitting the data, the model's prediction accuracy can be significantly im-proved. CEEMDAN-RMSE, LSTM's MAE, and $R^2$ decreased by 23.07%, 22.41%, and 39.72%, respectively, when compared to LSTM.
D. The ARIMA-DBO-CNN-LSTM model has a 64.02% reduction in RMSE, 77.78% reduction in MAE and 30.65% improvement in $R^2$ relative to the combined data processing model CEEMDAN-CNN-LSTM; and 84.71% reduction in RMSE, 84.78% reduction in MAE and 92.41% improvement in $R^2$ relative to the deep learning model DBO-LSTM. 84.78% and an increase in $R^2$ of 92.41%.
E. The DBO can effectively improve the prediction accuracy of the model, comparing the ARIMA-DBO-CNN-LSTM model with the ARIMA-CNN-LSTM model, the RMSE metric is reduced by 34.53%, MAE is reduced by 34% and $R^2$ is improved by 1.64%.
F. Using different models to predict different parts of the data can effectively im-prove the prediction accuracy and has better results than CEEMDAN decom-posed data. From the comparison between the derivative model of ARI-MA-DBO-CNN-LSTM model and the decomposed combined model, the deriva-tive model can reach above 0.95 in $R^2$ index, while the data decomposed com-bined model maintains between 0.7 and 0.8.
G. As can be seen in Fig. 7b, the ARIMA-DBO-CNN-LSTM model can predict the AQI of Jiaozuo City well in comparison with the single model and the combined model, and the scatter plot has the best aggregation effect.

Similar conclusions as Jiaozuo City can be drawn in the predictions of Beijing and Lanzhou, and the prediction pairs of Beijing and Lanzhou are shown in Figs. 8, 9 and 10. The combined prediction results of the three cities show that the ARIMA-DBO-CNN-LSTM model has the best prediction performance with an $R^2$ index of 0.989.

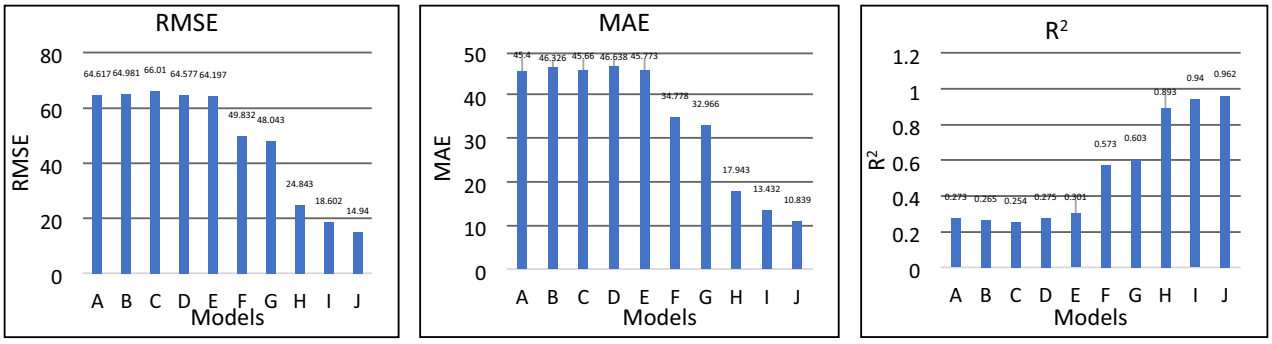
## Discussion and conclusion

Air pollution is an environmental problem faced worldwide, and effective AQI prediction can help in air pollution management. Traditional time series prediction models have large prediction errors in air quality prediction, which increasingly cannot meet the needs of current production and life. However, neural networks represented by LSTM have shown excellent prediction performance in time series prediction. In this paper, we predict AQI of four cities by building ARIMA-DBO-CNN-LSTM models. To verify the advantages of the models, the comparison models are not limited to the selection of derived models, and the current mainstream algorithmic models for air quality prediction are incorporated.

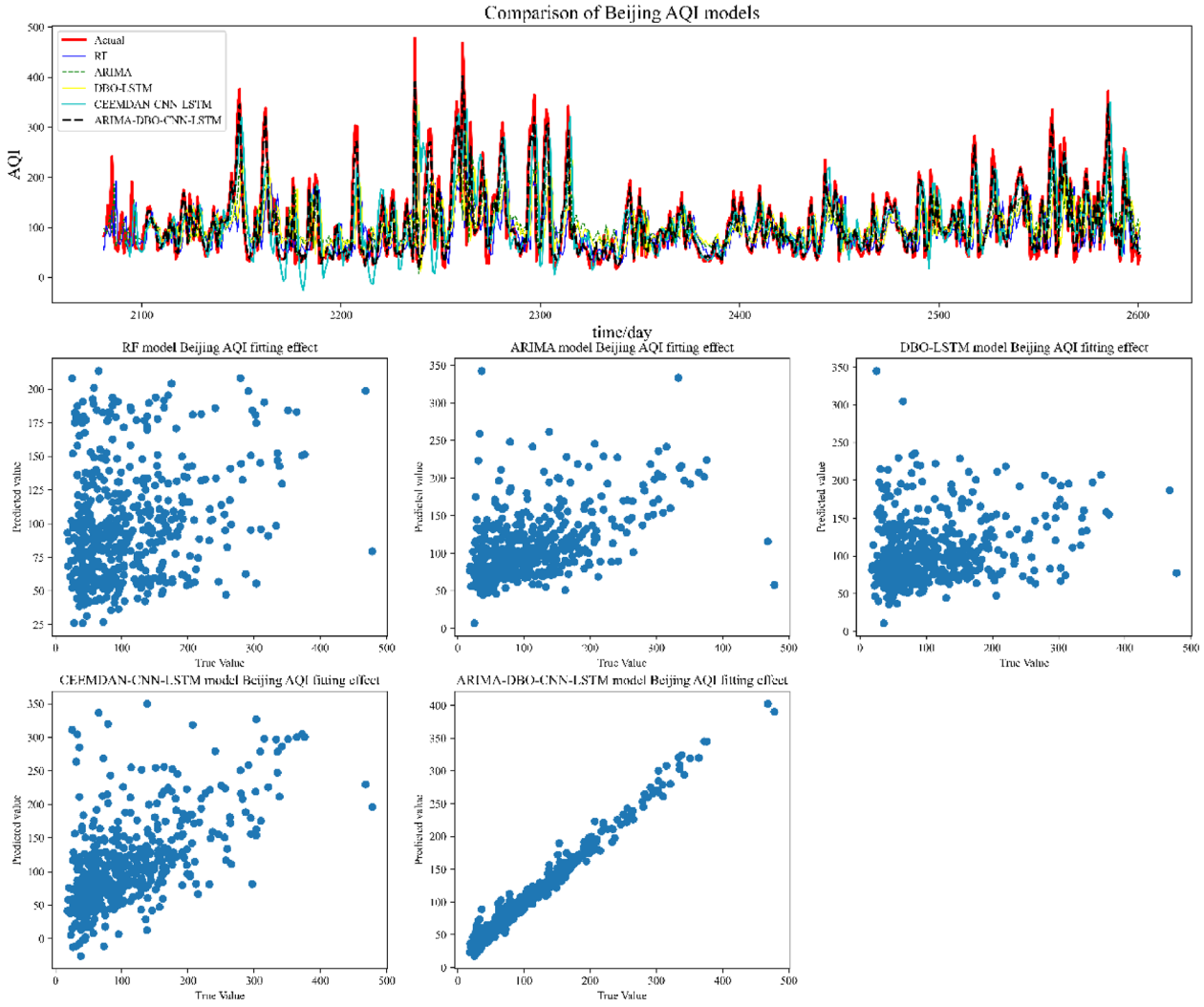
We used the AQI data detected in four cities, Beijing, Lanzhou, Jiaozuo and Guangzhou, to construct and analyze all models, and the experimental results show that the ARIMA-DBO-CNN-LSTM has good prediction effect on the test set. The experimental results show that the model proposed in this paper outperforms the comparison models in terms of root mean square error (RMSE), mean absolute error (MAE) and coefficient of determination (R2). The RMSE values of the four cities are 7.594, 14.94, 7.841 and 5.496; the MAE values are 5.285, 10.839, 5.12 and 3.77; the R2 values are 0.989, 0.962, 0.953 and 0.953, respectively. The ARIMA-DBO-CNN-LSTM model has higher prediction accuracy for the four cities and better adaptability. Among the four selected Chinese cities, Jiaozuo city has the best prediction accuracy performance with three evaluation indexes of 7.594, 5.285 and 0.989 for RMSE, MAE and R2, respectively.

The model proposed in this paper also has the following problems: (1) The proposed model consists of a combination of two models, and each group of models can only fit part of the model better, but not 100%, which will produce reaveraging. (2) There are many external factors that affect the air quality (AQI) index, such as various meteorological indicators and seasonal factors, which are not considered in this paper.

In the future, various influencing factors can be introduced into the model to improve the accuracy of the model. In conclusion, this study shows that our proposed model can achieve higher accuracy than traditional single models such as BiLSTM, while the method based on EMD decomposition and LightGBM integration has better performance than other decomposition integration methods. In addition, the model is not complicated to construct and is worthy to be applied in practice.

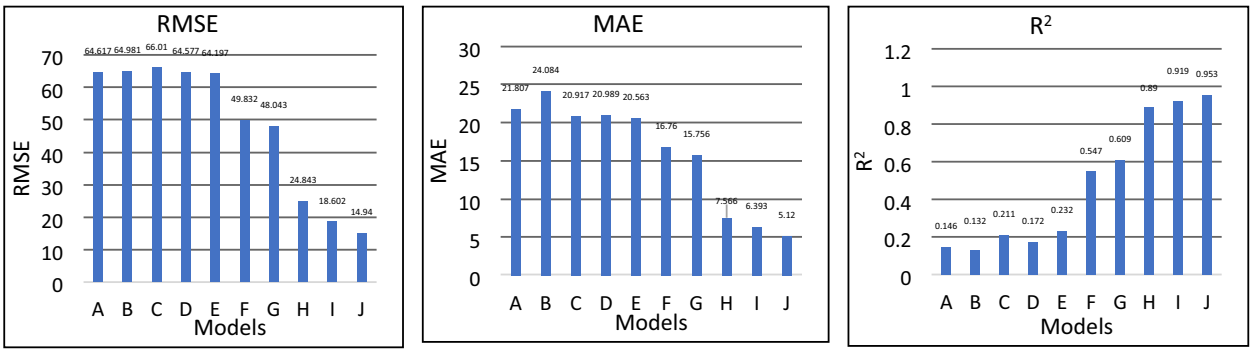



**(a)** Comparison of assessment indicators in Beijing City

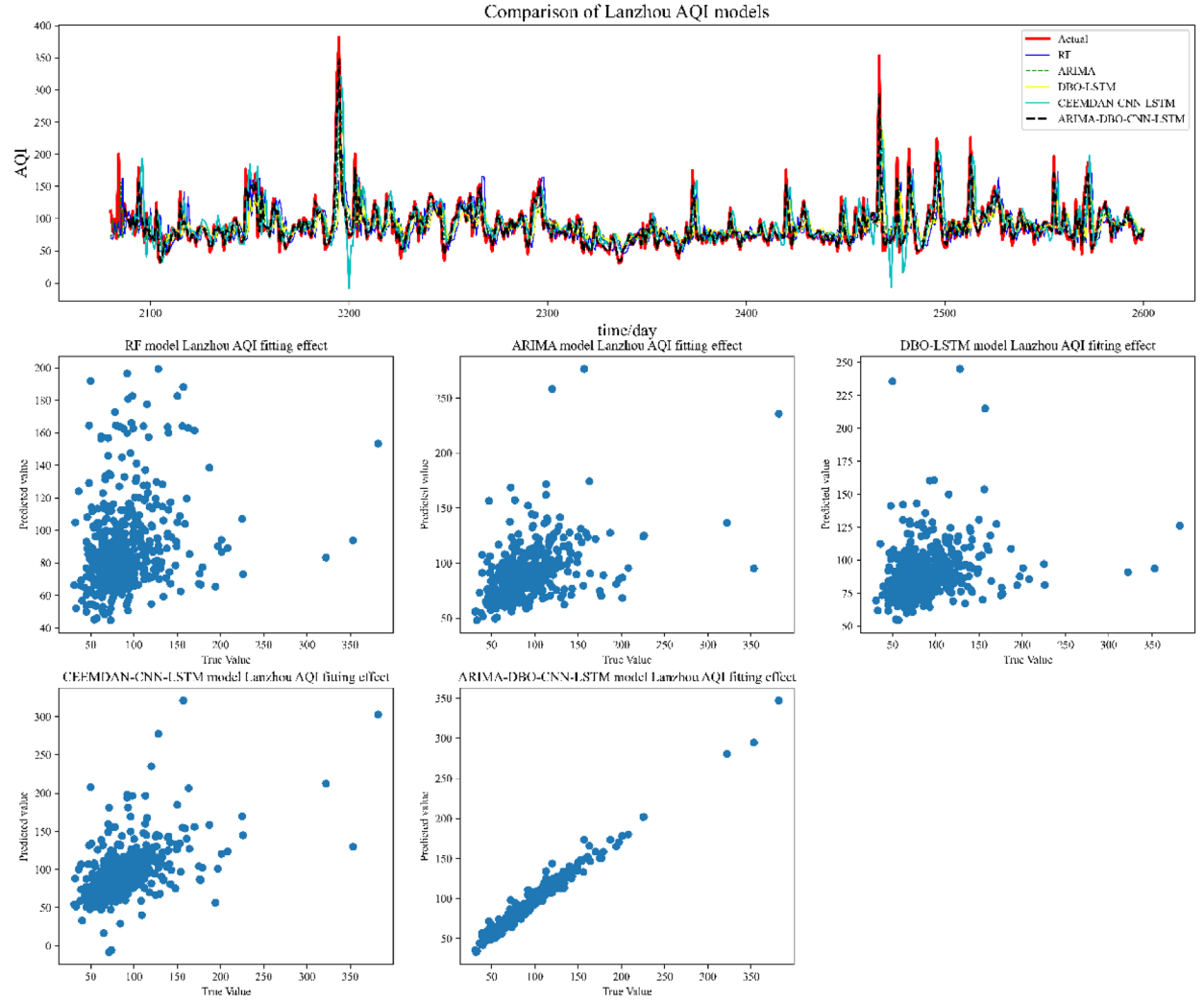



**(b)** Beijing City Model Scatter Comparison

**Figure 8.** Beijing City Forecast Comparison.

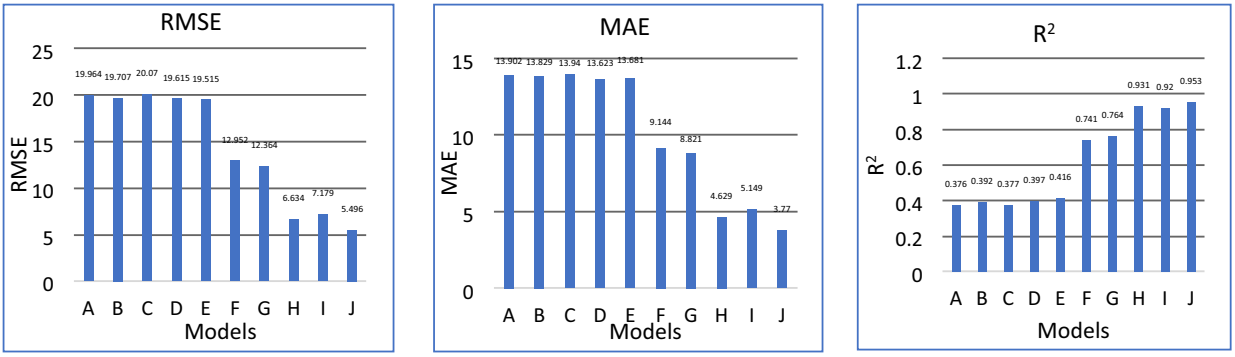



**(a)** Comparison of assessment indicators in Lanzhou City

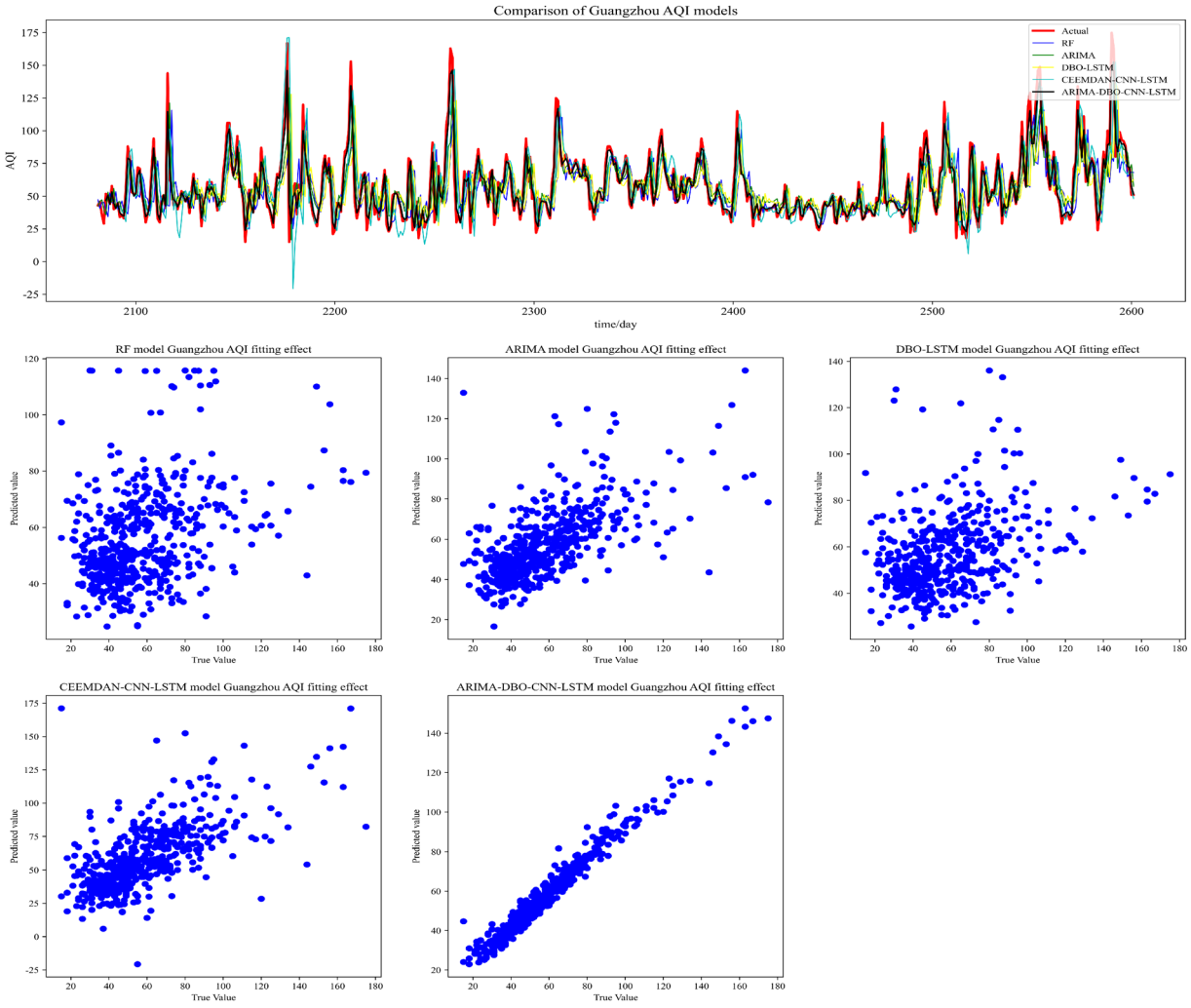



**(b)** Lanzhou City Model Scatter Comparison

**Figure 9.** Lanzhou City Forecast Comparison.




**(a)** Comparison of assessment indicators in Guangzhou City




**(b)** Guangzhou City Model Scatter Comparison

**Figure 10.** Guangzhou City Forecast Comparison.

## Data availability

The data that support the findings of this study are available from [Resource and Environment Science and Data Center of the Chinese Academy of Sciences (https://www.resdc.cn/Default.aspx)] but restrictions apply to the availability of these data, which were used under license for the current study, and so are not publicly available. Data are however available from the authors upon reasonable request and with permission of [Resource and Environment Science and Data Center of the Chinese Academy of Sciences].

Received: 23 March 2023; Accepted: 7 June 2023
Published online: 26 July 2023

## References

1. Suman, M. Air quality indices: A review of methods to interpret air quality status. *Mater. Today Proc.* **34**, 863–868 (2021).
2. Tanasa, I., Cazacu, M. & Sluser, B. Air quality integrated assessment: Environmental impacts, risks and human health hazards. *Appl. Sci.* **13**, 1222 (2023).
3. Zhang, F. Y., Xu, J. & Wang, L. Air quality, patterns and otolaryngology health effects of air pollutants in Beijing in 2013. *Aerosol Air Qual. Res.* **16**, 1464–1472 (2016).
4. Song, C. & Fu, X. S. Research on different weight combination in air quality forecasting models. *J. Clean. Product.* **261**, 121169 (2020).
5. Carbajal-Hernández, J. J., Sánchez-Fernández, L. P. & Carrasco-Ochoa, J. A. Assessment and prediction of air quality using fuzzy logic and autoregressive models. *Atmos. Environ.* **60**, 37–50 (2012).
6. Zhang, L. C., Tian, X. & Zhao, Y. H. Application of nonlinear land use regression models for ambient air pollutants and air quality index. *Atmos. Pollut. Res.* **12**, 101186 (2021).
7. Zhao, L., Li, Z. & Qu, L. Forecasting of Beijing PM(2.5) with a hybrid ARIMA model based on integrated AIC and improved GS fixed-order methods and seasonal decomposition. *Heliyon* **8**, e12239 (2022).
8. Zhou, W., Wu, X. & Ding, S. Predictive analysis of the air quality indicators in the Yangtze River Delta in China: An application of a novel seasonal grey model. *Sci. Total Environ* **748**, 141428 (2020).
9. Mehmood, K. *et al.* Predicting the quality of air with machine learning approaches: Current research priorities and future perspectives. *J. Clean. Product.* **379**, 134656 (2022).
10. Varghese, R. J. & Kumar, S. Machine learning model to predict the laminar burning velocities of H2/CO/CH4/CO2/N2/air mixtures at high pressure and temperature conditions. *Int. J. Hydrogen Energy* **45**, 3216–3232 (2020).
11. Zhang, W. X., Wu, Y. P. & Calautit, J. K. A review on occupancy prediction through machine learning for enhancing energy efficiency, air quality and thermal comfort in the built environment. *Renew. Sustain. Energy Rev.* **167**, 112704 (2022).
12. Rakholia, R., Le, Q. & Quoc Ho, B. Multi-output machine learning model for regional air pollution forecasting in Ho Chi Minh City, Vietnam. *Environ Int.* **173**, 107848 (2023).
13. Gu, Y. L., Li, B. H. & Meng, Q. G. Hybrid interpretable predictive machine learning model for air pollution prediction. *Neurocomputing* **468**, 123–136 (2022).
14. Maltare, N. N. & Vahora, S. air quality index prediction using machine learning for Ahmedabad city. *Digit. Chem. Eng* **7**, 100093 (2023).
15. Munir, S., Luo, Z. & Dixon, T. The impact of smart traffic interventions on roadside air quality employing machine learning approaches. *Transp. Res. Part D Transport Environ.* **110**, 103408 (2022).
16. Zhang, B., Rong, Y. & Yong, R. H. Deep learning for air pollutant concentration prediction: A review. *Atmos. Environ.* **290**, 119347 (2022).
17. Wu, C. L., He, H. D. & Song, R. F. A hybrid deep learning model for regional O(3) and NO(2) concentrations prediction based on spatiotemporal dependencies in air quality monitoring network. *Environ. Pollut.* **320**, 121075 (2023).
18. Jurado, X., Reiminger, N. & Benmoussa, M. Deep learning methods evaluation to predict air quality based on computational fluid dynamics. *Expert Syst. Appl.* **203**, 117294 (2022).
19. Zhang, K. J., Zhang, X. & Song, H. T. Air quality prediction model based on spatiotemporal data analysis and metalearning. *Wirel. Commun. Mob. Comput.* **2021**, 1–11 (2021).
20. Saez, M. & Barceló, M. A. Spatial prediction of air pollution levels using a hierarchical Bayesian spatiotemporal model in Catalonia, Spain. *Environ. Model. Softw.* **151**, 105369 (2022).
21. Kshirsagar, A. & Shah, M. Anatomization of air quality prediction using neural networks, regression and hybrid models. *J. Clean. Product.* **369**, 133383 (2022).
22. Zhang, J. & Li, S. Air quality index forecast in Beijing based on CNN-LSTM multi-model. *Chemosphere* **308**, 136180 (2022).
23. Vlachokostas, C., Achillas, C. & Chourdakis, E. Combining regression analysis and air quality modelling to predict benzene concentration levels. *Atmos. Environ.* **45**, 2585–2592 (2011).
24. Gunasekar, S., Joselin Retna Kumar, G. & Pius Agbulu, G. Air quality predictions in urban areas using hybrid ARIMA and metaheuristic LSTM. *Comput. Syst. Sci. Eng.* **43**, 1271–1284 (2022).
25. Wang, J. Y., Li, J. Z. & Wang, X. X. Air quality prediction using CT-LSTM. *Neural Comput. Appl.* **33**, 4779–4792 (2020).
26. Dai, H. B., Huang, G. Q. & Zeng, H. B. Haze risk assessment based on improved PCA-MEE and ISPO-LightGBM model. *Systems* **10**, 263 (2022).
27. Dai, H. B., Huang, G. Q. & Zeng, H. B. PM2.5 volatility prediction by XGBoost-MLP based on GARCH models. *J. Clean. Production* **356**, 131898 (2022).
28. Akilandeswari, P., Manoranjitham, T. & Kalaivani, J. Air quality prediction for sustainable development using LSTM with weighted distance grey wolf optimizer. *Soft Comput.* https://doi.org/10.1007/s00500-023-07997-1 (2023).
29. Du, P., Wang, J. Z. & Hao, Y. A novel hybrid model based on multi-objective Harris hawks optimization algorithm for daily PM2.5 and PM10 forecasting. *Appl. Soft Comput.* **96**, 106620 (2020).
30. Huang, Y., Xiang, Y. X. & Zhao, R. X. Air quality prediction using improved PSO-BP neural network. *IEEE Access* **8**, 99346–99353 (2020).
31. Sun, W. & Sun, J. Daily PM(2.5) concentration prediction based on principal component analysis and LSSVM optimized by cuckoo search algorithm. *J. Environ. Manag.* **188**, 144–152 (2017).
32. Abebe, M., Noh, Y. & Kang, Y.-J. Ship trajectory planning for collision avoidance using hybrid ARIMA-LSTM models. *Ocean Eng.* **256**, 111527 (2022).
33. Hasnain, A., Sheng, Y. & Hashmi, M. Z. Assessing the ambient air quality patterns associated to the COVID-19 outbreak in the Yangtze River Delta: A random forest approach. *Chemosphere* **314**, 137638 (2023).
34. Paulpandi, C., Chinnasamy, M. & Nagalingam Rajendiran, S. Multi-site air pollutant prediction using long short term memory. *Comput. Syst. Sci. Eng.* **43**, 1341–1355 (2022).
35. Xue, J. K. & Shen, B. Dung beetle optimizer: a new meta-heuristic algorithm for global optimization. *J. Supercomput.* **79**, 7305–7336 (2022).
36. Ji, C., Zhang, C. & Hua, L. A multi-scale evolutionary deep learning model based on CEEMDAN, improved whale optimization algorithm, regularized extreme learning machine and LSTM for AQI prediction. *Environ. Res.* **215**, 114228 (2022).
37. Xu, D., Zhang, Q. & Ding, Y. Application of a hybrid ARIMA-LSTM model based on the SPEI for drought forecasting. *Environ. Sci. Pollut. Res. Int.* **29**, 4128–4144 (2022).

## Acknowledgements

Thanks to the data support from the Resource and Environmental Science and Data Center of the Chinese Academy of Sciences.

## Author contributions

J.D. contributed to the conception of the study; J.D. and J.L. performed the experiment; J.D., Y.G. contributed significantly to analysis and manuscript preparation; J.D. performed the data analyses and wrote the manuscript; Y.G., J.L. and Z.Z. helped perform the analysis with constructive discussions.

## Competing interests

The authors declare no competing interests.

## Additional information

**Correspondence** and requests for materials should be addressed to Y.G.

**Reprints and permissions information** is available at www.nature.com/reprints.

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

# Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).
Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.
These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.
We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.
While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.
These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.
To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.
Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.
If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com