IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version Jan. 1, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.DOI

A Novel Combined Prediction Scheme Based on CNN and LSTM for Urban $PM_{2.5}$ Concentration

DONGMING QIN[1], JIAN YU[1], GUOJIAN ZOU[2], RUIHAN YONG[2], QIN ZHAO[2], AND BO ZHANG[2]

[1]Key Laboratory of Embedded Systems and Service Computing of Ministry of Education, Tongji University
[2]College of Information, Mechanical and Electrical Engineering, Shanghai Normal University

Corresponding authors: Qin Zhao (q_zhao@shnu.edu.cn) and Bo Zhang (zhangbo@shnu.edu.cn).

This work is supported in part by the National Natural Science Foundation of China (61572326, 61802258, 61702333), in part by the Natural Science Foundation of Shanghai (18ZR1428300), in part by the Shanghai Committee of Science and Technology (17070502800, 16JC1403000), and in part by the Opening Topic of Key Laboratory of Embedded Systems and Service Computing of Ministry of Education (ESSCKF 2016-01).

ABSTRACT Urban air pollutant concentration prediction is dealing with a surge of massive environmental monitoring data and complex changes in air pollutants. This requires effective prediction methods to improve prediction accuracy and prevent serious pollution incidents, thereby enhancing environmental management decision-making capacity. In this paper, a new pollutant concentration prediction method is proposed based on vast amounts of environmental data and deep learning techniques. The proposed method integrates big data using two kinds of deep networks. This method is based on a design that uses a Convolutional Neural Network as the base layer, automatically extracting features of input data. A Long Short Term Memory network is used for the output layer to consider the time dependence of pollutants. Our model consists of these two deep networks. With performance optimization, the model can predict future particulate matter ($PM_{2.5}$) concentrations as time series. Finally, the prediction results are compared with the results of numerical models. The applicability and advantages of the model are also analyzed. Experimental results show that it improves prediction performance compared with classic models.

INDEX TERMS Air pollution, machine learning, neural networks, numerical analysis, prediction methods

I. INTRODUCTION

Air pollution has attracted substantial attention regarding the daily life of people. It has negative impacts on human health and daily life during episodes of severe air pollution [1]. With the increase of sources and types of air pollutants, the complexity of pollutant concentration prediction has increased [2]. Therefore, it is necessary to use environmental monitoring data to more accurately predict urban air pollutant concentrations [3]. Conventional prediction methods, such as numerical analysis and machine learning, are widely used in this type of prediction [4]–[6]. However, several drawbacks of these methods have been recently identified as follows. First, numerical prediction methods are based on experience as summarized by historical data or the nature of pollutant change. Nevertheless, atmospheric conditions are too complex to assign a certain regular behavior because they are typically required regarding variables which are stochastically dependent [5, 32]. Thus, numerical methods cannot adequately consider atmospheric effects. Second, conventional machine learning methods simply consider the relationship among similar objects, e.g., adjacent monitoring stations, but ignore deep relationships, for instance, global spatial information or temporal changes in pollutants.

Recently, with increase in the application of deep learning to various fields [7]–[10], the study of urban air pollutant concentration prediction based on such learning has become prevalent in interdisciplinary research [11], [12]. Deep learning can use deep mining of massive environmental data and identify complex correlations between those data through effective training [10]. Compared with conventional prediction methods, deep learning-based methods can use massive amounts of environmental monitoring data in prediction systems. They can also consider spatiotemporal changes of pollutants and obtain the pollutant distribution. By gradually

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2897028, IEEE Access

training and adjusting the prediction model, it can achieve optimal performance and reduce prediction error.

Most deep learning-based methods use Long Short Term Memory (LSTM) [13] to make predictions [11], [12]. Thanks to its remarkable performance in time-series data processing, LSTM can handle time-related pollutant data well. However, this type of method ignores the spatial aspect of monitoring data. Obviously, change of pollutants is not only related to time but also to space. A pollutant at one location may diffuse to nearby places, so it is necessary to consider spatial information.

A Convolutional Neural Network (CNN) [14] has proven to be powerful in spatial data processing. It is widely used in image recognition [15], [16] . This type of method has also been used to predict urban pollutant concentration, typically by analyzing satellite images. Unfortunately, sometimes there are no image data but only abstract monitoring data, e.g., wind direction, temperature, and location. In fact, these data from monitoring stations are spatially relevant. Therefore, it is reasonable to use those data to predict urban pollution.

To overcome the drawbacks of existing methods, we propose a novel approach. Our motivation was to construct a prediction model that accounts for the complexity and variability of pollutants and eliminate dependence on the historical regularity of changing pollutants. More specifically, we combined CNN and LSTM to predict PM2.5 concentration. The rationale for this is as follows.

(1) CNN can learn and detect a specific type of feature at a spatial location in the input by the convolutional layer [17] . Given this advantage, we used the CNN to extract spatial features of inputs among monitoring stations, e.g., to learn the magnitude of spatial effects at different monitoring stations when there was air pollutant diffusion. Then, we could use the output as LSTM input in the next step.

(2) LSTM is a type of Recurrent Neural Network (RNN) [18] that has been proposed to predict future outputs using past inputs. LSTM has been shown to be well-suited for prediction based on time-series data, with better performance than RNN in dealing with exploding and vanishing gradient problems [19]–[21] . Therefore, we used LSTM to predict future air pollution concentrations by learning features contained in past air pollution concentration time-series data. For instance, we used LSTM to predict air pollutant concentration in the subsequent 3 hours by learning the past 24–72 hour tendency of that concentration.

The work is summarized as follows. (1) A prediction model is designed based on two deep neural networks, CNN and LSTM. A dynamic training method is used to train the model to extract features automatically until the best performance is obtained. (2) Feature extraction from the data is the primary purpose of the prediction system. This step is performed using CNN. The aim is to extract actual features from the input data and avoid unnecessary calculations that reduce the result accuracy. The CNN deals with the historical data using a series of convolutional and pooling operations. Then, the model enters the resultant feature maps into the LSTM. (3) The memory function of the LSTM network accounts for data dependence on time. The model time-series prediction results are therefore more accurate. (4) The elastic net (EN) [22] algorithm was used in the fine-tuning stage with the stochastic gradient descent method to carry out regularization constraints, adjust network weights, and avoid the over-fitting problem. Ultimately, network performance was adjusted to achieve optimal performance. Then, the $PM_{2.5}$ prediction result was compared with that of classic models with various measurements to demonstrate the effectiveness of the proposed method.

II. RELATED WORK

According to characteristics of the prediction methods used in relevant studies [4]–[6], air pollutant concentration prediction methods can be divided into conventional methods with non-deep learning and those based on deep learning.

Combined with methods of predicting pollutant concentrations in meteorology, environmental science, mathematics and computer science, the conventional prediction methods can be further divided into four types, predictions of empirical models based on historical data and statistical methods, predictions of probability models based on statistical and mathematical methods or models, predictions based on synthetic methods, and prediction models based on conventional machine learning.

Empirical models [23] do not analyze the process but count correlation data and determine the link between parameters and variables to obtain the corresponding relationship. For instance, the relationship between monthly mean pollutant concentration and other pollutant concentrations are established using an empirical statistical method [24], [25]. Historical data of pollutant concentrations can be modeled to predict changes in concentration through a chemical conversion model [6], [15], [24]. Probability models are based on statistical probability regularity and are combined with statistical or mathematical modeling methods. They are used to produce or select more precise prediction samples. Such research is based on experiment, and its predictions are established on probability and statistical models. Dong *et al.* used a hidden semi-Markov model and added temporal structures. Past meteorological measurements and the past historical observation concentration level of $PM_{2.5}$ were added to the training dataset, and corresponding Hidden Semi-Markov models (HSMMs) were trained for each concentration level. Prediction accuracy exceeded 24 hours. Balachandran *et al.* used Bayesian algorithms to investigate the effects of various pollutant sources on their concentrations [26]. Using the ensemble Kalman filter method to construct a regional pollution assimilation system is a classic synthetic approach combining numerical models with observations by using the optimal estimation method. Back propagation (BP) neural networks are frequently used in predictions based on conventional machine learning. Reference [27] considered Shenyang City as the monitoring center of a dataset as original data. The 120

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2897028, IEEE Access

sets of data in the autumn of 1999 and NO_x concentration data were selected as the training set, and meteorological data from 2000 used as the test set. The prediction model of pollutant concentration was established [27], and the predicted results were obtained and compared with observations.

The above methods effectively predict small-scale air pollutant concentration. However, large amounts of background data and the daily accumulation of air pollution-related data are not independent. They have time-dependent and spatial correlation. Conventional machine learning models do not have a deep network layer to limit excessive training costs and no coupling between the same layer of neurons, so they cannot solve the problem of time-dependent pollutant concentration.

Recently, the academic community has begun using deep neural networks for pollutant concentration prediction, because of the shortcomings of conventional prediction methods. Kuremoto *et al.* used a deep network composed of two restricted Boltzmann machines [11] to perform time-series prediction. By using the CATs benchmark [28] and original data, it has been proven that RBMs are superior to the ARIMA linear model. Ong *et al.* predicted air pollutant concentration with deep recurrent neural networks, which have been widely researched [11], [29], [30]. This shows that DRNN yields better results than RBMs under the same conditions [12].

However, massive input data should be processed and features and correlations extracted. Then, time-series features should be extracted because pollution constitutes dependent on past historical data. Owing to its unique structure, CNN can use convolution kernels to convolve features of neighboring regions to obtain the features' spatial correlation [14]. The extraction of spatial correlations helps determine the effects of air quality and meteorological conditions of neighboring cities on target cities. LSTM is superior in processing time-series data because the concentration of air pollutants is time-dependent, and the historical concentration affects the future concentration. Details of the CNN and LSTM are given in Section III.

We combined CNN and LSTM as the prediction model. The CNN convolutional layer was used as the basis for extracting features, and its shareable local weights reduced network complexity. Compared with RNNs, LSTM can solve long-term dependence problems and can better predict pollutant concentration in a time series. Therefore, these two networks were combined and had the ability to extract features in both spatial and temporal dimensions. Further, spatial and temporal effects can be introduced in the prediction system to obtain better predictions.

III. THE PREDICTION MODEL BASED ON DEEP LEARNING

A. TIME SERIES PREDICTION

We constructed a prediction model combining the CNN and LSTM. The CNN was used to eliminate data redundancy and acquire the features. LSTM was used to extract the time-series features because current pollutant concentrations are affected by the past pollutant concentration and past meteorological factors. Following is a description of the time-series prediction.

Given a set of time series times $R = \{r_1, r_2, ..., r_T\}$, the target pollutant concentration is denoted as $C = \{C_1, C_2, ..., C_T\}$, and the information of the other factors are denoted as $F = \{f_1, f_2, ..., f_T\}$, so there is a relationship denoted as $C \cup F = R$.

Given the time t, $z_1 = (t+1, t+2, ..., t+N)$ is the target prediction time series. O_{z_1} and P_{z_1} are the observations and predictions of the target pollutant concentration, respectively, in the time series and $z_1 \in [1, T]$, $O_{z_1} \subset R$. $z_2 = (t, t-1, ..., t-D)$ is a time series before time t used for predicting the target pollutant concentration for the following N hours. $O^c_{z_2}$ and $O^f_{z_2}$ are the observations of the target pollutant and other factors, respectively, in the previous D hours. Similarly, there are constraints denoted as $z_2 \in [1, T]$ and $O^c_{z_2} \cup O^f_{z_2} \subset R$.

The prediction uses the root mean square equation (RMSE) to assess its error:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(O_i - P_i)^2}{N}}, \tag{1}$$

where N is the prediction duration, O_i is the observed value and P_i is the predicted value of the of the target pollutant. The smaller the RMSE is, the better the performance of the model.

B. THE PREDICTION MODEL

Deep learning was proposed by Dechter in 1986 [31]. It is a machine learning process that can carry out a series of training for sample data through unsupervised training methods and obtain a deep network structure. We exploited CNN characteristics that compress and extract important features of input data, along with the unique structure of LSTM designed for the time-series problem.

The prediction model uses the CNN as the base layer, compressing and extracting features using its convolutional and pooling layers. The output of the CNN layer is the input of a higher layer, LSTM, for the time series prediction. The model is shown in Fig. 1.

Meteorological factors and pollutant concentration from the past are included in the prediction model as input. They are converted to several two-dimensional matrices with time series. Then, these matrices are input to the CNN network to extract the features. The output is used as input to the LSTM. The fully connected layer is used to decode the LSTM output and obtain the final prediction result.

Assuming that the model has δ layers, μ is the layer that is currently being trained. x_i is a set of input data, and y_i is a set of output data without decoding of the fully connected layer. i is the dynamically changing time, and C^μ_i, P^μ_i, L^μ_i are the outputs of the convolutional layers, pooling layers and LSTM layers, respectively. The deep learning model for pollutant

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2897028, IEEE Access

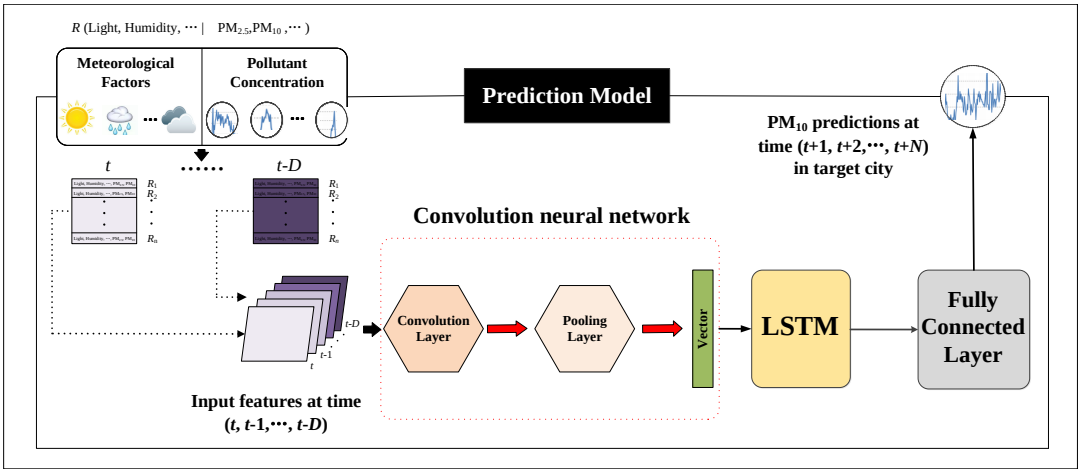


FIGURE 1. Prediction Model

concentration prediction in this paper can be expressed by the following.

$$\begin{aligned} C_i^{\mu} &= g(u^{\mathrm{T}} x_i), \qquad \mu = 2 \\ P_i^{\mu} &= g(v^{\mathrm{T}} C_i), \qquad 2 < \mu < \delta - 3 \\ L_i^{\mu} &= g(w^{\mathrm{T}} P_i + d^{\mathrm{T}} L_i), \qquad 3 < \mu < \delta - 2 \\ y_i^{\mu} &= f(\xi^{\mathrm{T}} L_i), \qquad \mu = \delta - 2 \end{aligned} \tag{2}$$

where u, v, w, d, ξ are the weight matrices of the prediction model: u is the weight matrix of the input layer to the convolution layer; v is the weight matrix of the convolution layer to the pooling layer; w is the weight matrix of the pooling layer to the LSTM layer; d is the weight matrix of the information transfer between the LSTM layer internal neurons; ξ is the weight matrix of LSTM to the fully connected layer. The resulting y_i is decoded by the fully connected layer and translated into the pollutant concentration value. The parameters used for model training are shown in Table 1.

C. TRANING PROCESS

1) Traning models

The prediction model consists of two parts, so the training process is divided into two steps.

Step 1. Traning for the CNN

The CNN can automatically learn the features of input data, so it is unnecessary to extract data features before training. The input features are converted to two-dimensional matrices. η is the number of layers trained and m is the feature map. The characteristic graph of the upper layer output of the convolution layer is studied using the convolution kernel k of the convolution layer. The output feature graph is obtained using the activation function. The output feature maps of the previous convolutional layer are studied by the convolutional kernel k of the current convolutional layer and produces its feature maps through the activation function. i, j are subscripts of the feature maps.

$$m_j^{\eta} = f(\sum_{i \in M} m_i^{\eta-1} \times k_{ij}^{\eta} + b_j^{\eta}) \tag{3}$$

After the feature map is convoluted in CNN, there are N feature maps as input to the pooling layer, which outputs N features with contractible sizes.

$$m_j^{\eta} = f(\beta_j^{\eta} down(m_j^{\eta-1}) + b_j^{\eta}) \tag{4}$$

where β and b are respectively the multiplicative bias and additive bias respectively of the output maps. $down$ is the down-sampling function. N feature maps are expanded into N one-dimensional vectors, and the output pollutant concentration value is obtained using the full connection layer decoding.

Training of the CNN is shown in Fig. 2. The two-dimensional matrix input of this stage includes the following features: $\{PM_{2.5}$ $concentration$, $temperature$, $wind$ $speed$, $wind$ $direction$, $humidity$, $precipitation$, $other$ $pollutant$ $concentration\}$. Prediction accuracy is measured using the RMSE. By using the back-propagation algorithm and considering the pooling layer as a factor, the weights of the convolution layer are updated based on all values. Then, the network prediction performance is optimized, and error between the predicted and observed values is reduced.

This training stage compresses the two-dimensional input matrix and attains the actual data features, so the network can accurately translate the input data into pollutant concentration values and map the input to output. When the network meets expectations, the first stage of the network training is stopped, and the next stage of training begins.

Step 2. Training for the global model

In Fig. 3, the output of the CNN's last pooling layer is input to the LSTM layer. The two-dimensional input matrices are compressed and the features extracted are then converted

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2897028, IEEE Access

TABLE 1. Model Parameters

Parameters	Value
Training method (for prediction model)	Stochastic gradient descent
Kernel size of convolution layer	5×5
Kernel size of pooling layer	2×2
Number of convolution layers	1
Number of pooling layers	1
Number of convolution layer parameters	$5 \times 5 \times 6$
Number of LSTM layers	1
Number of LSTM nodes	256
Number of fully connected layers	2
Number of first fully connected layer nodes	128
Number of second fully connected layer nodes	64
Learning rate	0.005
Batch size	64

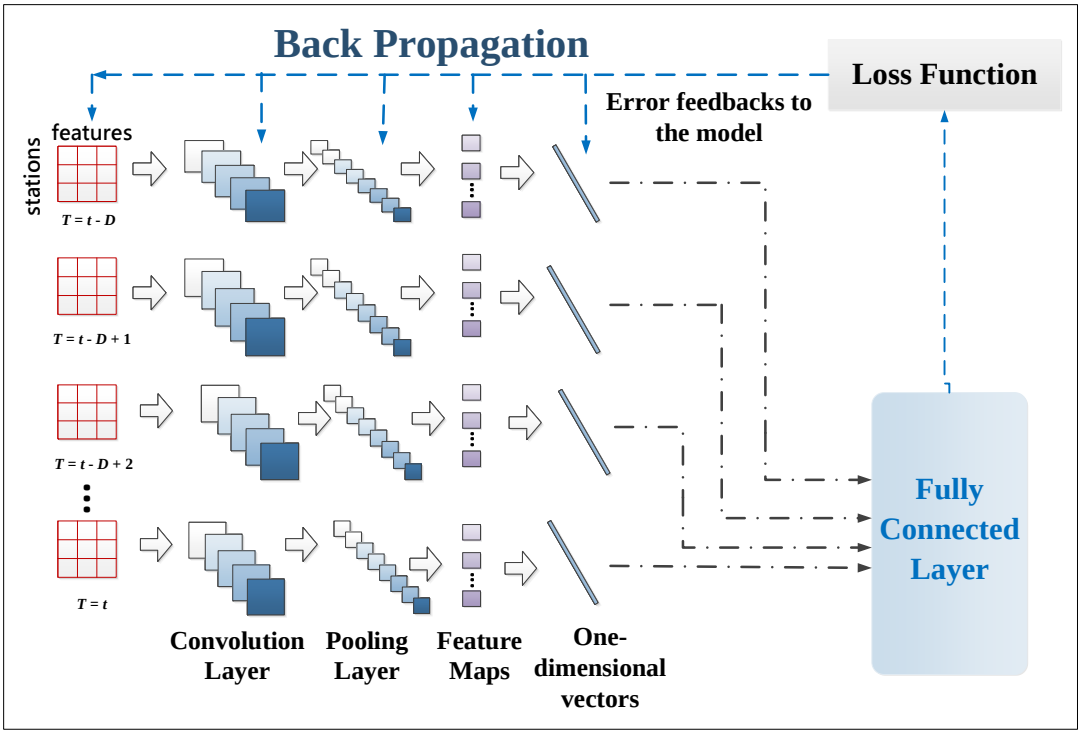


FIGURE 2. CNN structure of the model

to highly concentrated one-dimensional vectors with time-series characteristics. Values of $O^c_{(t,\ldots,t-D)}$ and $O^f_{(t,\ldots,t-D)}$ for D hours before time t are model input, and the prediction target is the hourly $PM_{2.5}$ concentration value (D and N are the set time windows) for N hours after t. x is the input and represents the dynamic time series. W is a weight matrix. h is the hidden layer information and b is the bias. The following formulas are used to represent the training process for LSTM:

i. The LSTM first selectively forgets some past $PM_{2.5}$ data information and other factors:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{5}$$

ii. Deciding what new information to store in the unit state, the new information originates from two parts. The "input threshold" sigmoid layer determines the updated information and the tanh layer creates a new candidate value vector:

$$\begin{aligned} i_t &= \sigma(W_i[h_{t-1}, x_t] + b_i) \\ C'_t &= \tanh(W_C[h_{t-1}, x_t] + b_t) \end{aligned} \tag{6}$$

iii. Updating the previous state:

$$C_t = f_t \times C_{t-1} + i_t \times C'_t \tag{7}$$

iv. Finally, determining the output information, i.e., the predicted $PM_{2.5}$ concentration:

$$\begin{aligned} o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o) \\ h_t &= o_t \times \tanh(C_t) \end{aligned} \tag{8}$$

Details of the prediction model framework are shown in Fig. 3. The final LSTM output prediction result is decoded

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2897028, IEEE Access

by the CNN's fully connected layer. Considering that deep neural networks are prone to fitting problems during training, in the fine tuning stage, we used the stochastic gradient descent algorithm for the model. The EN algorithm, which combines the advantages of ridge regression and lasso to carry out L_1 and L_2 regularization constraints, was used in this stage. The error function was used to update the gradient of all weights and bias values of the network, using error backpropagation. The loss function and gradients of all weights were calculated, and bias values of the network were updated using the error backpropagation until the expected network was obtained. This avoided the over-fitting problem.

LSTM adds the time-series prediction function to the model. Its inputs are one-dimensional vectors with real data features after CNN convolution and pooling. Therefore, complex and unnecessary calculations are avoided, and the time dependence of pollutants is considered.

2) Regularization and objective function

To solve the over-fitting problem in deep networks, the EN algorithm is used for regularization constraints, so that the objective function in the training fine-tuning stage can achieve the minimum. Compared with other regression algorithms, the advantages of the EN algorithm are listed in Table 2.

The advantages of EN were experimentally confirmed in [17], choosing the following as the objective function.

$$E(\varphi) = \sqrt{\frac{\sum_{i=1}^{N}(O_i - P_i)^2}{N}} + \frac{\lambda}{2}((1-\zeta)|\varphi| + \zeta\varphi^{\mathrm{T}}\varphi) \quad (9)$$

The objective function of the network was established as the sum of the RMSE and regular term. In (9), the first half is the RMSE and N is the prediction horizon. In the subsequent part, λ is a nonnegative hyper-parameter. φ is the collection of the weights as in Section III-B, $\varphi = \{u, v, w, d, \xi\}$. ζ is a parameter that controls the ratio of L_1, L_2 penalty, $\zeta \in (0, 1)$.

IV. EXPERIMENTS AND RESULTS

We chose Shanghai as the target city. The dataset used contains data from three years (2015 to 2017), which were collected manually. Pollutant and the meteorological information of 2015 and 2016 were used as the training set. The 2017 data were used as the test set. The parameters used to train the prediction model are listed in Table 1. In this dataset, there are 14 observation stations, in the target city and neighboring cities. For each of the 14 stations, we used features $\{PM_{2.5}\ concentration, temperature, wind\ speed, wind\ direction, humidity, precipitation, other\ pollutant\ concentration\}$ as input to our model. The other pollutant features were CO, CO_2, NO, NO_2, SO_2, and PM_{10}. The input consisted of 72 hours of past data. The output was a predicted sequence of 24 $PM_{2.5}$ values in time series for the target city. Prediction experiments for each model were run 10 times. The results were the means of RMSEs over all runs, and the maximum number of epochs was 100. Details of the dataset are listed in Table 3.

In the experiment, the RMSE and correlation coefficient ($Corr$) were used as measurements. The RMSE equation is the same as (1) in section III-A. $Corr$ is expressed as

$$Corr = \frac{Cov(O, P)}{\sqrt{Var[O] \times Var[P]}}, \quad (10)$$

where O is the observed value and P is the predicted value. $Cov(O, P)$ is the covariance of O and P. $Var[O]$ and $Var[P]$ represent the variances of O and P, respectively.

Fig. 4 shows the RMSE variation of each model. Fitting trends corresponding to different epochs during the training period are also shown. To demonstrate the advantage of the proposed model, we chose three classic models, the BP neural network, RNN, and LSTM. Each model used the same dataset as the proposed method, and their fitting trends are shown in Fig. 4.

In Fig. 4, panels (a), (b), (c), (d) and (e) represent the fitting trends of five different models. All were chosen as fitting trend figures for the same numbers of epochs, which were 10, 30, 50, 70, 90 and 100. During the training period of each model, the predicted value moved toward the actual one. The prediction of the proposed method (CNN+LSTM) best fits the actual results.

To reveal the prediction performances of BP, RNN, CNN, LSTM and our proposed method, we list the results in Table 4. We see that our model had the best performance of all the above. By comparing the CNN-based method with RNN and BP based methods, we see that the CNN could improve the pollutant concentration prediction $Corr$. Also, the $Corr$ of the CNN-alone based method is larger than our proposed method at 0.01. This means that both methods can achieve good performance in predicting the trend of air pollutant concentration. However, the RMSE of the CNN-alone based method is much worse than our proposed method and LSTM-alone based method, which means that CNN performance is poor in dealing with long-term sequence prediction. Further, comparing the LSTM-alone based method with our proposed model, the experimental results show that the LSTM prediction performance without considering spatial correlation features is poorer than that of CNN+LSTM. This indicates that the prediction performance of the proposed model can be improved by adding time-series feature information based on the correlation of spatial features. Such a result reveals that we can add the CNN to LSTM to improve the prediction performance of air pollutant spatiotemporal data. The BP, CNN, RNN and LSTM neural networks were unsuitable for spatiotemporal sequence prediction problems because prediction accuracy was poor over time. To show the superior prediction accuracy of the proposed model, the final RMSE and $Corr$ values of each model are listed in Table 4.

V. CONCLUSION

We exploited massive amounts of environmental data and proposed a fusion network based on the CNN and LSTM.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2897028, IEEE Access

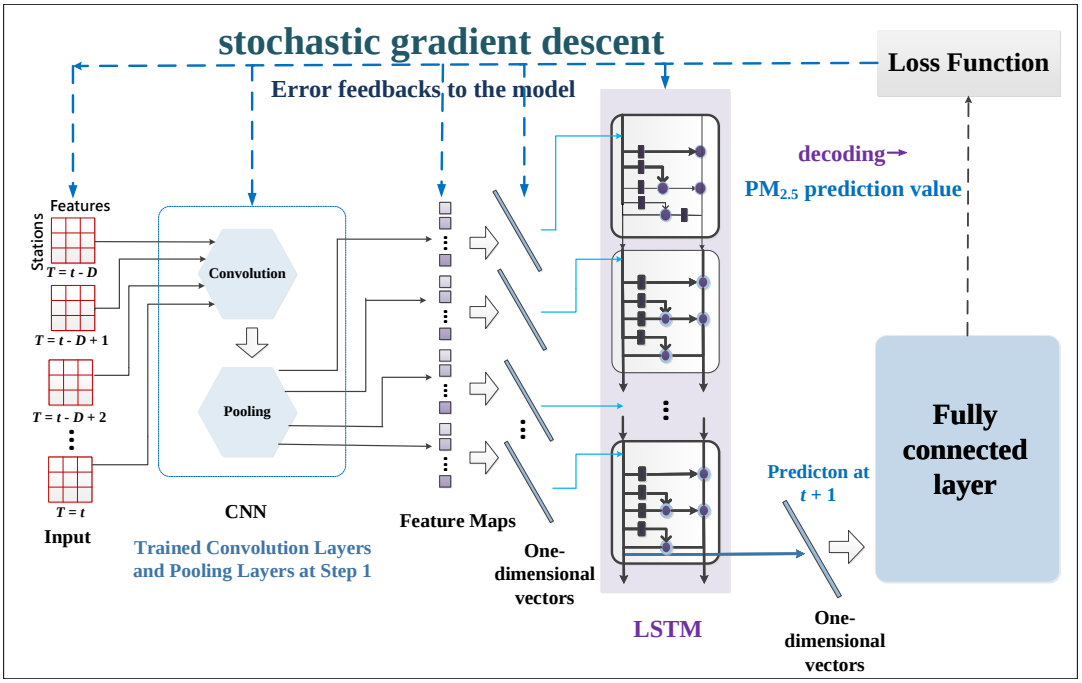


FIGURE 3. The framework of the prediction model

TABLE 2. Comparison of algorithms

Algorithm	Advantages	Disadvantages
Linear Regression	• Easy to implement	• Suitable for low dimension • Unsuitable for multi-collinearity
Ridge Regression	• Using L_2 penalty • Realistic regression coefficients	• Unable to select variables • Unable to shrink parameters to zero
Lasso	• Using L_1 penalty • Able to select variables • Able to shrink parameters to zero	• Inconsistent • Unable to perform group selection for a set of highly relevant variables
Elastic Net	• Combination of ridge regression and lasso • Variable selection based on sparsity • Maintaining regularization and stability of ridge regression	• Optimal ratio of L_1, L_2 penalty have to be obtained from multiple experiments.

TABLE 3. Details of Experiment Dataset

Parameters	Value
Datasets span	2015-2017
Training set	2015-2016
Test set	2017
Prediction horizon	24 hours
Past data	72 hours
Size of the input matrices (stations*features)	14×7
Maximum epochs	100

TABLE 4. RMSEs and Corr values of each model

Model	RMSE (epochs=100)	*Corr* (epochs=100)
BP	22.37	0.92
CNN	30.66	0.98
RNN	30.66	0.89
LSTM	17.95	0.95
CNN+LSTM (proposed)	**14.3**	**0.97**

The CNN was the basis of the model and used to extract spatial features of air pollutants. LSTM was the top of the model and used to extract time series features for the input. The advantages of the proposed method are summarized as follows.

1) The input data are compressed to eliminate redundancy and obtain actual features using the CNN. Spatial correlation between the data was determined after convolution and pooling. Additionally, because the CNNs use shared weights, the complexity of the prediction model was reduced.
2) LSTM addressed the time series problem because the pollutants have time dependence. The trained CNN and untrained LSTM were trained and fine-tuned together to obtain the final model. Regularization was used to avoid over-fitting.

In general, the proposed model is suitable for processing data from multiple monitoring sites in a single city as input to a time series. It can incorporate the interaction of mul-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2897028, IEEE Access

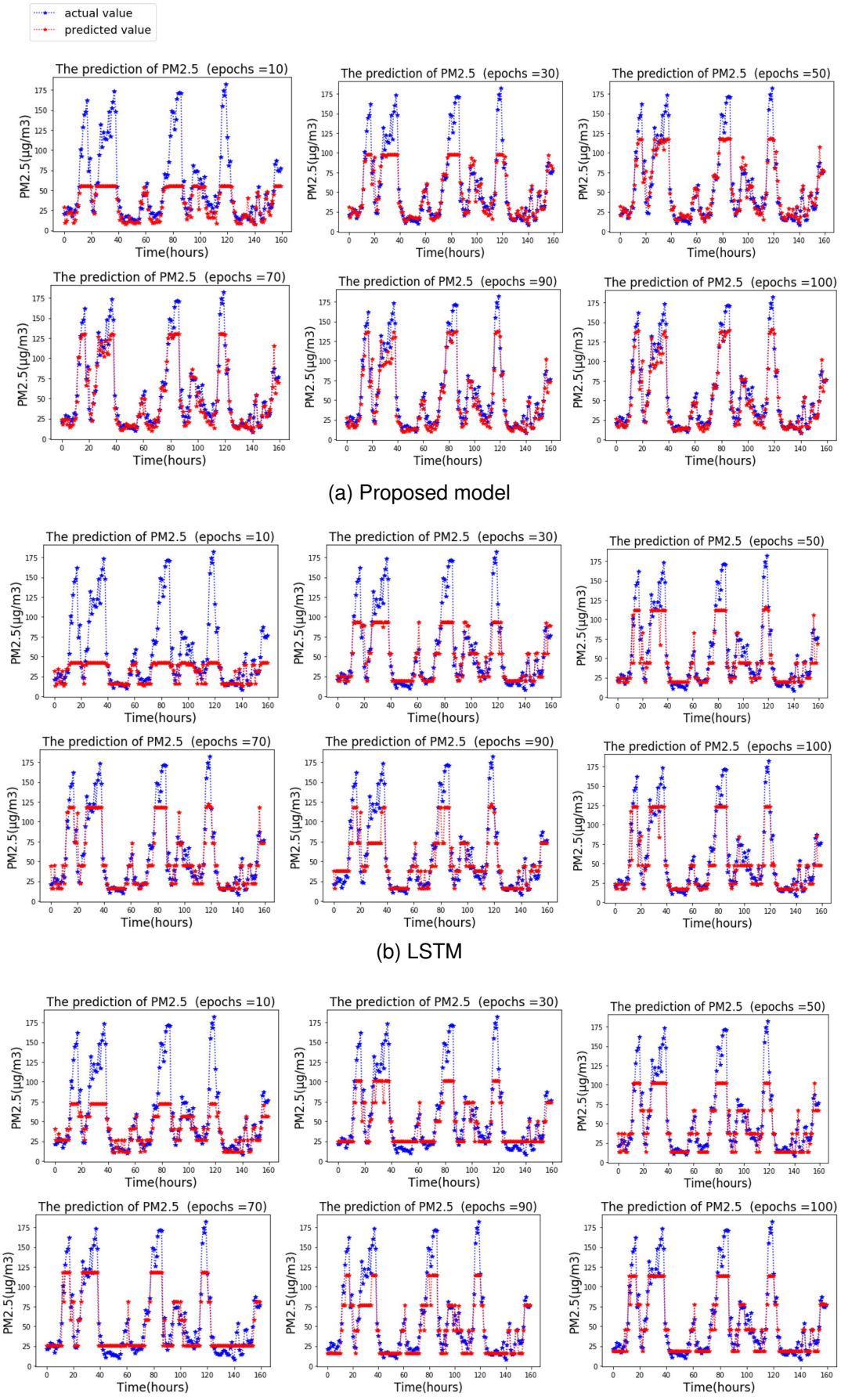
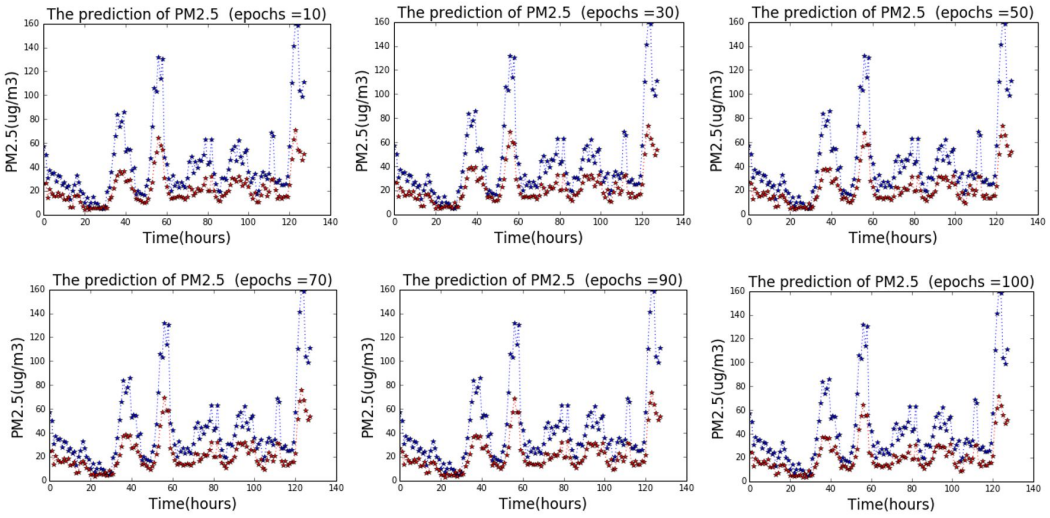
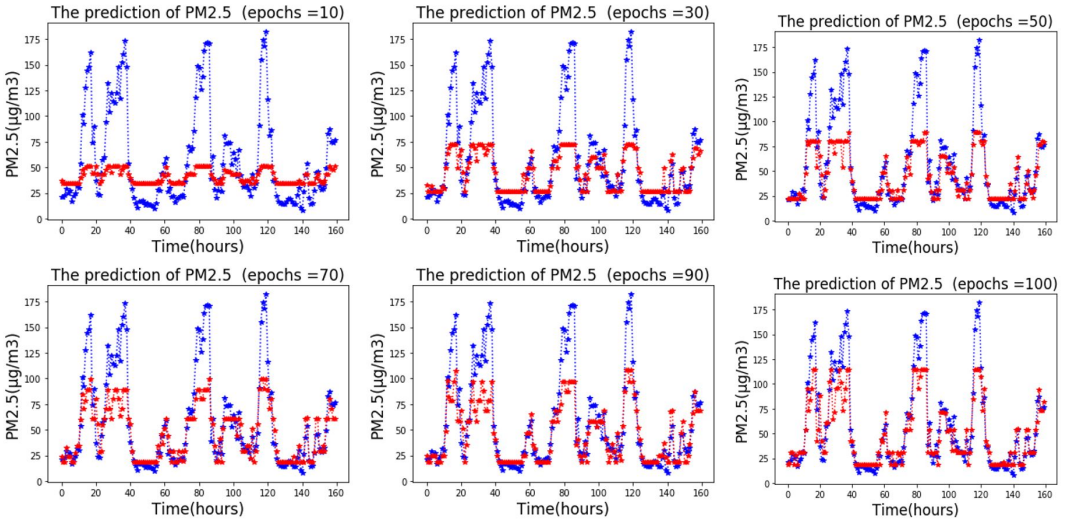


FIGURE 4. Fitting trends of models

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2897028, IEEE Access



(d) CNN



(e) BP

FIGURE 4. Fitting trends of models (continued)

tiple sites and temporal dependence of air pollutants in the prediction system. There are some limitations of proposed work: (1) the training data of our model is used from multiple sites, (2) the work is given based on only one city (Shanghai) and we want to collect more monitoring data from other cities to verify the generalization of our work, and (3) more factors, e.g., geomorphic conditions, need to be taken into account in our future work. We can thereby better determine the regularity of air pollutant data and achieve more accurate prediction results.

REFERENCES

[1] C. Sun, M. E. Kahn, and S. Zheng, "Self-protection investment exacerbates air pollution exposure inequality in urban China," Ecological Economics, vol. 131, pp. 468–474, 2017.

[2] Q. Zhang et al., "Transboundary health impacts of transported global air pollution and international trade," Nature, vol. 543, no. 7647, p. 705, 2017.

[3] L. Gharibvand et al., "The association between ambient fine particulate air pollution and lung cancer incidence: results from the AHSMOG-2 study," Environmental health perspectives, vol. 125, no. 3, p. 378, 2017.

[4] A. Lee, A. Szpiro, S. Y. Kim, and L. Sheppard, "Impact of preferential sampling on exposure prediction and health effect inference in the context of air pollution epidemiology," Environmetrics, vol. 26, no. 4, pp. 255–267, 2015.

[5] S. Park et al., "Predicting PM10 concentration in Seoul metropolitan subway stations using artificial neural network (ANN)," Journal of hazardous materials, vol. 341, pp. 75–82, 2018.

[6] I. Djalalova, L. Delle Monache, and J. Wilczak, "PM 2.5 analog forecast and Kalman filter post-processing for the Community Multiscale Air Quality (CMAQ) model," Atmospheric Environment, vol. 108, pp. 76–87, 2015.

[7] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: a deep learning approach," IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 2, pp. 865–873, 2015.

[8] D.L. Yamins and J.J. DiCarlo, "Using goal-driven deep learning models to understand sensory cortex," Nature neuroscience, vol. 19, no. 3, p. 356,

2016.

[9] L. Deng and D. Yu, "Deep learning: methods and applications," Foundations and Trends® in Signal Processing, vol. 7, no. 3–4, pp. 197–387, 2014.

[10] J. Schmidhuber, "Deep learning in neural networks: An overview," Neural networks, vol. 61, pp. 85–117, 2015.

[11] T. Kuremoto, S. Kimura, K. Kobayashi, and M. Obayashi, "Time series forecasting using a deep belief network with restricted Boltzmann machines," Neurocomputing, vol. 137, pp. 47–56, 2014.

[12] B.T. Ong, K. Sugiura, and K. Zettsu, "Dynamically pre-trained deep recurrent neural networks using environmental monitoring data for predicting PM2. 5," Neural Computing and Applications, vol. 27, no. 6, pp. 1553–1566, 2016.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in Advances in neural information processing systems, pp. 91–99, 2015.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," IEEE transactions on pattern analysis and machine intelligence, vol. 37, no. 9, pp. 1904–1916, 2015.

[17] Y. Kim, "Convolutional neural networks for sentence classification," arXiv preprint arXiv:1408.5882, 2014.

[18] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," nature, vol. 323, no. 6088, p. 533, 1986.

[19] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 961–971.

[20] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on LSTM recurrent neural network," IEEE Transactions on Smart Grid, 2017.

[21] M. Sundermeyer, H. Ney, and R. Schlüter, "From feedforward to recurrent LSTM neural networks for language modeling," IEEE Transactions on Audio, Speech, and Language Processing, vol. 23, no. 3, pp. 517–529, 2015.

[22] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 67, no. 2, pp. 301–320, 2005.

[23] K. R. Felzer, "The Empirical Model," Appendix Q in the Uniform Calfiornia Earthquake Rupture Forecast Version 3 (UCERF 3), The Time Indepdendent Model, USGS Open File Report 2013-1165, 2013.

[24] Y. H. Xie, M. M. Zhang, L. Yang, and H. D. Zhang, "Predicting urban PM 2.5 concentration in China using support vector regression," Computer Engineering & Design, pp. 3106–3111, 2015.

[25] H. Guo, Z. Y. Fu, Y. J. Xiong, and S. Y. Shi, "Analysis of the weather conditions for a case of heavy pollution in Beijing," Meteorol Monthly, vol. 33, no. 6, pp. 32–36, 2007.

[26] S. Balachandran, H. H. Chang, J. E. Pachon, H. A. Holmes, J. A. Mulholland, and A. G. Russell, "Bayesian-based ensemble source apportionment of PM2. 5," Environmental science & technology, vol. 47, no. 23, pp. 13511–13518, 2013.

[27] J. Wang, X. M. Hu, and L. X. Zheng, "Study on forecasting of air pollution based on BP model," Research of Environmental Sciences, vol. 15, no. 5, pp. 62–64, 2002.

[28] A. Lendasse, E. Oja, O. Simula, and M. Verleysen, Time series prediction competition: The CATS benchmark. Elsevier, 2007.

[29] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu, "Advances in optimizing recurrent networks," in Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, 2013, pp. 8624–8628.

[30] J. Fan, Q. Li, J. Hou, X. Feng, H. Karimian, and S. Lin, "A Spatiotemporal Prediction Framework for Air Pollution Based on Deep RNN," ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 4, p. 15, 2017.

[31] R. Dechter, "Learning While Searching in Constraint-satisfaction-problems," in Proceedings of the Fifth AAAI National Conference on Artificial Intelligence, Philadelphia, Pennsylvania, 1986, pp. 178–183.

[32] Corani, Giorgio, and M. Scanagatta. "Air pollution prediction via multi-label classification." Environmental Modelling & Software 80(2016):259-264.