Hindawi
Advances in Meteorology
Volume 2018, Article ID 3506394, 15 pages
https://doi.org/10.1155/2018/3506394

Research Article

A Multiple Kernel Learning Approach for Air Quality Prediction

Hong Zheng[1], **Haibin Li**[1], **Xingjian Lu**[1,2] **and Tong Ruan**[1]

[1]*Information Engineering and Computer Science College, East China University of Science and Technology, Shanghai 200237, China*
[2]*Smart City Collaborative Innovation Center, Shanghai Jiao Tong University, Shanghai 200240, China*

Correspondence should be addressed to Hong Zheng; zhenghong@ecust.edu.cn

Received 15 November 2017; Revised 29 January 2018; Accepted 7 May 2018; Published 12 June 2018

Academic Editor: Ilan Levy

Air quality prediction is an important research issue due to the increasing impact of air pollution on the urban environment. However, existing methods often fail to forecast high-polluting air conditions, which is precisely what should be highlighted. In this paper, a novel multiple kernel learning (MKL) model that embodies the characteristics of ensemble learning, kernel learning, and representative learning is proposed to forecast the near future air quality (AQ). The centered alignment approach is used for learning kernels, and a boosting approach is used to determine the proper number of kernels. To demonstrate the performance of the proposed MKL model, its performance is compared to that of classical autoregressive integrated moving average (ARIMA) model; widely used parametric models like random forest (RF) and support vector machine (SVM); popular neural network models like multiple layer perceptron (MLP); and long short-term memory neural network. Datasets acquired from a coastal city Hong Kong and an inland city Beijing are used to train and validate all the models. Experiments show that the MKL model outperforms the other models. Moreover, the MKL model has better forecast ability for high health risk category AQ.

1. Introduction

With the development of the economy and society all over the world, most metropolitan cities are experiencing elevated concentrations of ground-level air pollutants, especially in fast developing countries like India and China. Exposure to air pollution can affect everyone, but it can be particularly harmful to people with a heart disease or a lung condition, elderly people, and children. Studies show that long-term exposure to fine particulate air pollution or traffic-related air pollution is associated with environmental-cause mortality, even at concentration ranges well below the standard annual mean limit value [1, 2]. Therefore, building an early warning system, which provides precise forecast and also alerts health alarm to local inhabitants will provide valuable information to protect humans from damage by air pollution.

Currently, three major approaches are used to forecast real-time air quality: simple empirical approaches, advanced physically based approaches, and machine learning approaches.

Simple empirical approaches like persistence method and climatology method are based on assumptions or hypothesis; that is, thresholds of forecasted meteorological variables can indicate future pollution level [3]. They are computationally fast but have low accuracy and are primarily used as references by other methods. Advanced physically based approaches like chemical transport models (CTMs) simulate the formation and accumulation of air pollutants by a solution of the conservation equations and transformation relationships among the mass of various chemical species and physical states. They can provide valuable insights for understanding pollutant diffusion mechanisms. But they are computationally expensive, demanding reliable meteorological predictions, and highly relevant to a high level of expertise [4].

Machine learning methods are computationally fast and cost-effective and can provide promising prediction accuracy. Various machine learning methods have been applied to predict the air quality. Widely used methods include classical autoregressive moving average (ARMA) methods like the autoregressive integrated moving average

(ARIMA) [5], support vector machine (SVM) methods like the support vector classifier (SVC) [6, 7], ensemble methods like the random forest (RF) [8, 9], artificial neural network (ANN) methods like the multiple layer perceptron (MLP) [10, 11], and deep learning methods like the long short-term memory neural network (LSTM NN) [12, 13].

Among the models mentioned above, ARIMA is a time series model and is often used as a baseline model. The performance of the SVM model is often hinged on the appropriate choice of the kernel. A kernel in SVM introduces nonlinearity into the problem by mapping new input data implicitly into a Hilbert space where it may then be linearly separable [14]. Neural network models, especially deep neural networks, can automatically learn the representations from raw data, but it takes a long time and a large volume of data to train a well-behaved network.

Multiple kernel learning (MKL) is proposed as an alternative to cross validation, feature selection, metric learning, and ensemble methods. MKL refers to using multiple kernels instead of a single one; most of the algorithms which make use of the kernel tricks can take the advantage of MKL, such as SVM and kernel ridge regression (KRR). In MKL, feature combination and classifier training are done simultaneously, and different data formats can be used in the same formulation. In addition, the inherent kernel trick of combining linear kernels and nonlinear kernels in MKL makes it more promising in solving fusing information problems. There is a significant amount of work in the literature for combining multiple kernels [15, 16]. Various applications indicate that performance gains can be achieved by linear and nonlinear kernel combinations using MKL methods [17–19].

In this paper, a novel multiple kernel learning-based air quality prediction approach that can inherently capture the characteristics of the heterogeneous time, meteorology, and air pollutant data is proposed. Real datasets from a coastal city Hong Kong and an inland city Beijing are used to demonstrate the effectiveness the proposed approach. Comprehensive comparison experiments with ARIMA, RF, SVCs, MLP, and LSTM are conducted. Though some of the algorithms can automatically learn the representative features of the data, pretraining featuring engineering is still necessary and will significantly affect the models' performance. In addition, hyperparameter tuning is critical for all the parametric models. Therefore, in this paper, special attention is paid to the feature engineering and parameter tuning process. The methodologies applied to Hong Kong and Beijing datasets are similar. Therefore, Hong Kong is used for demonstration in most of the paper. The main contributions of this paper are as follows:

(1) A multiple kernel learning approach is introduced into the domain of air quality prediction for the first time. Multiscale predictions over the next 1, 3, 6, 9, and 12 hours' air quality of an inland city Beijing and a coastal city Hong Kong are presented.

(2) The proposed method can effectively capture the air quality features from the hybrid time, meteorology, and air pollutant data. The experimental results demonstrated the advantages of this approach over some of the widely used models, especially in the prediction of severe air pollution conditions.

The rest of the paper is organized as follows: Section 2 presents the methodology of the multiple kernel learning algorithm; data preparation is introduced in Section 3; in Section 4, extensive experimentation results and necessary discussions are presented; and Section 5 concludes this paper.

2. Methodology

While classical kernel-based classifiers such as SVCs are based on a single kernel, in practice, it is often desirable to base classifiers on combinations of multiple kernels since data points typically can be due to multiple heterogeneous sources. A kernel implicitly represents a notion of similarity for the data, and different kernels will accommodate different nonlinear mappings, and MKL provides a way to combine different ideas of similarity. Using a specific kernel may be a source of bias, and MKL provides a way to select optimal kernels and parameters from a larger set of kernels. In the air quality prediction case, the source data are coming from different modalities. Therefore, in the paper, instead of using just a single kernel which is usually more suitable for the homogeneous data source, multiple kernels are combined, and the classical and empirically successful support vector classifier is used as the base learner. The detailed introduction of the kernel support vector machine is given in Appendix A. In this section, the multiple kernel learning approach is described first, and then, the centered alignment method is introduced for learning kernels.

2.1. Multiple Kernel Learning. MKL is conceptually similar to single kernel learning. In other words, single kernel leaning is a special case of MKL. In MKL, the final kernel is learnt as a combination (linear or nonlinear) of many base kernels from the data itself:

$$\kappa_\eta\left(x_i, x_j\right) = f_\eta\left(\left\{\kappa_m\left(x_i^m, x_j^m\right)\right\}_{m=1}^{P}\middle|\eta\right), \tag{1}$$

where f_η: $\mathbb{R}^P \to \mathbb{R}$ is the combination function, κ_m is the kernel function, m is the dimensionality of the corresponding feature representation, and η parameterizes the combination function.

It is also possible to integrate η into the kernel functions where it is optimized during training.

$$\kappa_\eta\left(x_i, x_j\right) = f_\eta\left(\left\{\kappa_m\left(x_i^m, x_j^m\middle|\eta\right)\right\}_{m=1}^{P}\right). \tag{2}$$

Most of the existing MKL algorithms fall into the first category and try to combine predefined kernels in an optimal way. Commonly used kernels are linear, polynomial, radial basis function (RBF), and sigmoid.

Input: dataset: $(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(n)}, y^{(n)})$, n samples
Output: decision function of MKSVC
Start
First, get the kernel coefficients by optimizing the single kernel-base learners $(\kappa_m(x_i, x_i))$
Second, get the weight of each kernel by the centered kernel alignment algorithm (η)
Third, get the number of kernels by boosting approach (P)
Fourth, get the combined optimized kernel $\kappa_\eta(x_i, x_j) = \sum_{m=1}^{P} \eta_m \kappa_m(x_i, x_i)$
Then, use SVC as the base learner and optimize it with a general optimizing algorithm
Return $f_m(x) = \sum_{i=1}^{N} \alpha_i y_i \kappa_\eta(x_i, x_j) + b$
Stop

Algorithm 1: MKSVC.

$$\kappa\left(x,\ x^i\right) = \begin{cases} \left(x^T \cdot x^i\right) & \text{linear} \\ \left(x^T \cdot x^i + 1\right)^d & \text{polynomial} \\ \exp\left(-\gamma \left\|x - x^i\right\|^2\right) & \text{RBF} \\ \tan \mathrm{h}\left(\gamma x \cdot x^i\right) & \text{sigmoidal.} \end{cases} \tag{3}$$

The kernels can be combined in different ways, and each has its own combination parameter characteristics. Generally, linear combination methods are used, and they fall into two basic categories: unweighted sum (i.e., using sum or mean of the kernels as the combined kernel) and weighted sum. In the weighted sum case, the combination function is linearly parameterized:

$$\kappa_\eta\left(x_i, x_j\right) = f_\eta\left(\left\{\kappa_m\left(x_i^m, x_j^m\right)\right\}_{m=1}^{P} \middle| \eta\right) = \sum_{m=1}^{P} \eta_m \kappa_m\left(x_i^m, x_j^m\right), \tag{4}$$

where η denotes the kernel weights. Different versions of this approach differ in the way they put restrictions on η: the linear sum has arbitrary real value η_m and the conic sum requires η_m to be positive, while η sums to 1 for the convex sum.

The conic sum and convex sum are special cases of the linear sum, but the former two are used more often because the relative importance of the combined kernels can be extracted by looking at the kernel weights. Furthermore, the kernel weights of the conic and convex sum correspond to scaling the feature spaces when they are nonnegative [20].

In this paper, the conic sum restriction used as the convex sum is a special case of the conic sum. The resulting decision function of the multiple kernel support vector classifier (MKSVC) is defined as

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i \sum_{m=1}^{P} \eta_m \kappa_m\left(x_i^m, x_j^m\right) + b, \quad \text{subject to} \quad \eta \in \mathbb{R}_+^P. \tag{5}$$

There are four important parameters: the number of kernels (P), the inner kernel coefficients of each kernel, features to use for each kernel (x_i^m), and the weight (η_m) of each kernel. In this paper, the inner kernel coefficients are obtained by optimizing the single kernel-based learners. η is obtained by the centered alignment approach proposed in [32]. P is obtained through the boosting approach by iteratively adding a new kernel until the performance stops improving (the kernels are added based on the weights learned by the centered alignment approach, kernel with higher weight first). As with the features used by each kernel, for simplicity, the canonical multiple kernel learning approach is used, namely, one kernel combination for all feature representations. The pseudo code of the MKSVC is described in Algorithm 1.

2.2. Centered Alignment Method for Learning Kernels. Centered alignment is used as a similarity measure between kernels or kernel matrices. Given p kernels matrices $\mathrm{K}_1, \mathrm{K}_2, \ldots, \mathrm{K}_p$, centered kernel alignment learns a linear combination of kernels resulting in a combined kernel matrix:

$$\mathrm{K}_{c\mu} = \sum_{q=1}^{p} \mu_q \mathrm{K}_{cq}, \tag{6}$$

where p is the number of kernels, μ_q is the centered kernel weight, and K_{cq} is the centered kernel:

$$\mathrm{K}_{cq} = \left(\mathbf{I} - \frac{11^T}{m}\right) \mathrm{K}_q \left(\mathbf{I} - \frac{11^T}{m}\right), \tag{7}$$

where $\mathbf{I}$ is the identity matrix, $1 \in \mathbb{R}^{m \times 1}$ denotes the vector with all entries equal to one, and K_q is the original kernel matrix.

The alignment between two kernel functions K and K′ is defined by

$$\widehat{\rho}\left(\mathrm{K}, \mathrm{K}'\right) = \frac{\left\langle \mathrm{K}_c, \mathrm{K}_{\mathbf{c}}' \right\rangle_F}{\left\|\mathrm{K}_c\right\|_F \left\|\mathrm{K}_{\mathbf{c}}'\right\|_F}, \tag{8}$$

where K_c and $\mathrm{K}_{\mathbf{c}}'$ are the centered kernels of K and K′ and $\langle \cdot, \cdot \rangle_F$ denotes the Frobenius product and $\| \cdot \|_F$ the Frobenius norm defined by

$$\begin{aligned} &\forall \mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times m}, \\ &\langle \mathbf{A}, \mathbf{B} \rangle_F = \mathrm{Tr}\left[\mathbf{A}^T \mathbf{B}\right], \\ &\|\mathbf{A}\|_F = \sqrt{\langle \mathbf{A}, \mathbf{A}_F \rangle_F}, \end{aligned} \tag{9}$$

and $\widehat{\rho}(\mathrm{K}, \mathrm{K}') \in [0, 1]$ by definition.

Using the independent alignment-based algorithm proposed in [32], the alignment between each kernel matrix K_q and the target K_Y ($\mathrm{K}_Y = \mathbf{y}\mathbf{y}^T$, $\mathbf{y}$ is the labels) can be

computed independently by using the training samples and the centered kernel weight can be chosen proportional to that alignment. Thus, the resulting kernel matrix is defined by

$$K_\mu \propto \sum_{q=1}^{p} \widehat{\rho}\left(K_q, K_Y\right) K_q = \frac{1}{\left\|K_Y\right\|_F} \sum_{q=1}^{p} \frac{\langle K_q, K_Y \rangle_F}{\left\|K_q\right\|_F} K_q. \quad (10)$$

3. Data Preparation

In this paper, two datasets are used: one is from Hong Kong, a coastal city, whose air condition is relatively good, and the other is from an inland city, Beijing, whose air condition is relatively poor. Dataset of HK contains two years' hourly meteorology data and pollutant data between 1 February 2013 and 31 January 2015 collected from HK's Sha Tin air quality monitoring station [21] and weather forecast station [22]. Dataset of Beijing contains five years' hourly PM2.5 data and meteorology data between 1 January 2010 and 31 December 2014 collected from UCI machine learning repository [23].

3.1. Prediction Target and Performance Metric

3.1.1. Prediction Target. The prediction targets in this paper are the air quality health index (AQHI) in Hong Kong and the PM2.5 individual air quality level (IAQL) in Beijing. AQHI and IAQL are scales designed to help understand the impact of air quality on health. Unlike air quality concentrations, these air quality indices provide the public with advice on how to protect their health during air quality levels associated with low, moderate, high, and very high health risks. They also provide advice on how to improve air quality by proposing behavioral change to reduce the environmental footprint [24, 25].

For any given hour, the AQHI is calculated from the sum of the percentage excess risk of daily hospital admissions attributing to the 3-hour moving average concentrations of four criteria air pollutants: ozone (O_3), nitrogen dioxide (NO_2), sulphur dioxide (SO_2), and particulate matter (PM) (respirable suspended particulates (RSP or PM10) or fine suspended particulates (FSP or PM2.5), whichever poses a higher health risk).

The IAQL is classified based on the individual air quality index (IAQI) which is calculated according to a formula published by China' Ministry of Environmental Protection (MEP) [26]. The highest IAQI among pollutants SO_2, NO_2, O_3, carbon monoxide (CO), PM2.5, and PM10 at a given time is called the primary or dominant pollutant and is chosen for the overall AQI value. In China, PM2.5 is the primary pollutant most of the time; therefore, its IAQI is usually the overall AQI.

The detailed information of calculating AQHI and IAQI is given in Appendix B. These indices are health protection tools used to make decisions to reduce short-term exposure to air pollution by adjusting activity levels during increased levels of air pollution. Table 1 shows the health risks with corresponding air quality classifications.

Table 1: Air quality classifications and health risk.

Health risk	Low	Moderate	High	Very high	Serious
Hong Kong (AQHI)	1–3	4–6	7	8–10	10+
Beijing (IAQL)	1–2	3	4	5	6

Table 2: Air pollutant samples.

Date	Hour	Station	FSP	NO_2	NO_x	O_3	RSP	SO_2
1/1/2014	1	SHATIN	91	131	266	N.A.	114	18
1/1/2014	2	SHATIN	88	124	262	3	110	14
1/1/2014	3	SHATIN	86	114	225	2	107	13
1/1/2014	4	SHATIN	85	107	197	3	104	15

3.1.2. Performance Metric. In this paper, accuracy, mean square error (mse), weighted precision (wp), weighted recall (wr), and weighted f1-score (wf) are used to evaluate the effectiveness of all the algorithms. The precision (P) is calculated by the formula TP/(TP + FP) where TP is the number of correct predictions and FP is the number of incorrect predictions. Recall (R) is the proportion of instances classified as a given class divided by the actual total in that class. F1-score is a harmonic average of precision and recall [27].

For accuracy and mse,

$$\begin{aligned} \text{accuracy}(y, \widehat{y}) &= \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}} 1(y_i = \widehat{y}_i), \\ \text{mse}(y, \widehat{y}) &= \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}} (y_i - \widehat{y}_i)^2, \end{aligned} \quad (11)$$

where $\widehat{y}_i$ is the predicted value of the ith sample and y_i is the corresponding true value.

For wp, wr, and wf

$$\begin{aligned} \text{wp} &= \frac{1}{\sum_{l \in L} |\widehat{y}_l|} \sum_{l \in L} |\widehat{y}_l| P(y_l, \widehat{y}_l), \\ \text{wr} &= \frac{1}{\sum_{l \in L} |\widehat{y}_l|} \sum_{l \in L} |\widehat{y}_l| R(y_l, \widehat{y}_l), \\ \text{wf} &= \frac{1}{\sum_{l \in L} |\widehat{y}_l|} \sum_{l \in L} |\widehat{y}_l| F_1(y_l, \widehat{y}_l), \end{aligned} \quad (12)$$

where y is the set of predicted (sample, classes) pairs, $\widehat{y}$ is the set of true (sample, classes) pairs, L is the set of classes, and y_l is the subset of y with classes l; similarly, $\widehat{y}_l$ is the subset of $\widehat{y}$. $P(y_l, \widehat{y}_l) = |y_l \cap \widehat{y}_l| / |y_l|$ and $R(y_l, \widehat{y}_l) = |y_l \cap \widehat{y}_l| / |\widehat{y}_l|$ (conventions vary on handling $\widehat{y}_l = \phi$; this implementation uses $R(y_l, \widehat{y}_l) = 0$, and similar for $P(y_l, \widehat{y}_l)$). $F_1(y_l, \widehat{y}_l) = (2 \times (P \times R)) / (P + R)$.

3.2. Featured Data.

Take dataset of HK for example. Following air pollutant data features are contained: FSP, NO_2, NO_x, O_3, RSP, and SO_2 (unit of measurement of all the air pollutants is $\mu g/m^3$). Air pollutant data samples are shown in Table 2.

Table 3: Meteorological samples.

Local time in Sha Tin	T	P0	P1	δP	H	WD	WP	dew
29.01.2015 02:00	15.2	763.4	764.5	1.0	73	Wind blowing from the east	3	10.3
29.01.2015 01:00	15.6	763.9	765.1	0.4	77	Calm, no wind	0	11.5

T, air temperature (degrees Celsius) at 2 meters height above the Earth's surface; P0, atmospheric pressure at weather station level (millimeters of mercury); P1, atmospheric pressure reduced to mean sea level (millimeters of mercury); δP, pressure tendency, changes in atmospheric in the last three hours; H, relative humidity (%) at a height of 2 meters above the Earth's surface; WD, mean wind direction (compass points) at a height of 10–12 meters above the Earth's surface over the 10-minute period immediately preceding the observation; WP, mean wind speed at a height of 10–12 meters above the Earth's surface over the 10-minute period immediately preceding the observation (meters per second); dew, dew point at 2 meters height above the Earth's surface (degrees Celsius).

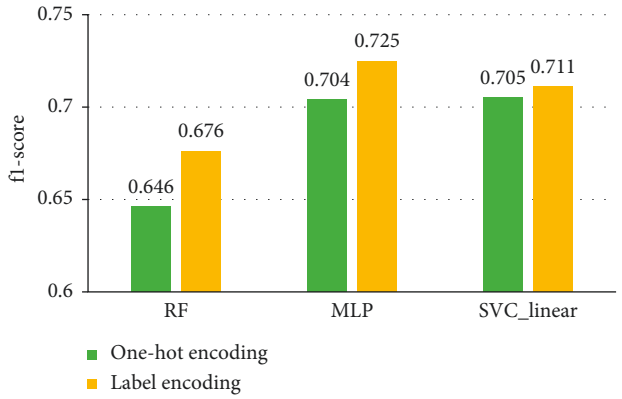

Figure 1: Comparison of one-hot encoding and label encoding over wind direction.

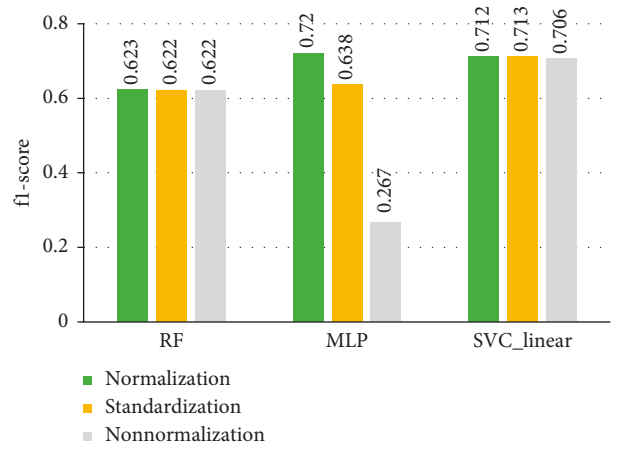

Figure 2: Comparison of with and without normalization.

Following meteorology data features are contained: T, P0, P1, δP, H, WD, WP, and dew. Meteorological samples are shown in Table 3.

Following time stamp features are contained: month, the day of the week (week), the day of the month (day), and the hour of the day (hour). There may be a yearly trend of the air quality, but we just have limited years of data, so "year" is not included in the feature set.

3.3. Feature Engineering

3.3.1. Feature Transformation

(1) Encoding Wind Direction. Among the data obtained, the wind direction is nonnumeric (i.e., "east," "east-southeast"). It has to be converted to numerical value so that the algorithms can make use of. One-hot encoding (e.g., "east" is encoded as [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]) and label encoding (e.g., "east" is encoded as 1, "south" is encoded as "2" etc.) were tried in this paper. Figure 1 shows the forecast performances of RF, MLP, and SVC_linear (SVC with linear kernel) algorithms when the wind direction was encoded by one-hot encoding and label encoding, respectively, and the parameters of the algorithms stayed unchanged. From the figure, it is obvious that label encoding is superior over one-hot encoding on the dataset. Therefore, in this paper, the wind direction was label encoded.

(2) Missing Data Imputation. Linear interpolation was used in the paper to interpolate the missing values in the two datasets.

$$V_t = \frac{V_s + (V_e - V_s)}{n+1}, \tag{13}$$

where V_t denotes the missing value at time t and n is the time gap between interval (V_s, V_e).

(3) Data Normalization. Normalization or standardization of either input or target variables tends to make the training process better behaved. Normalization scales the feature values in the range [0,1]:

$$V = \frac{V - V_{\min}}{V_{\max} - V_{\min}}. \tag{14}$$

Standardization transforms the feature values to have zero mean and unit variance:

$$V = \frac{V - \mu}{\sigma}. \tag{15}$$

To see whether normalization or standardization helps, both of them were tried and compared with the one without any processing. Again, RF, MLP, and SVC_linear were used as the validation algorithms. Results are shown in Figure 2. The figure shows that, generally, models benefit from normalization or standardization, especially for the neural network model. Normalization is slightly better than standardization. Therefore, in this paper, the data were normalized.

3.3.2. Feature Selection. Take Hong Kong for example. The source dataset contains 18 features, and they are as follows:

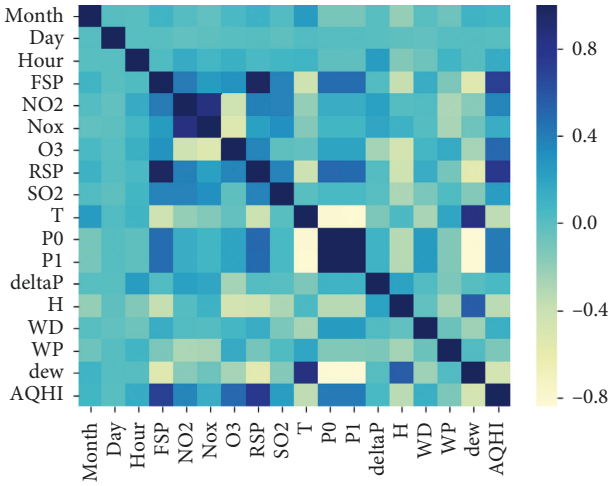

Figure 3: Spearman correlation between features.

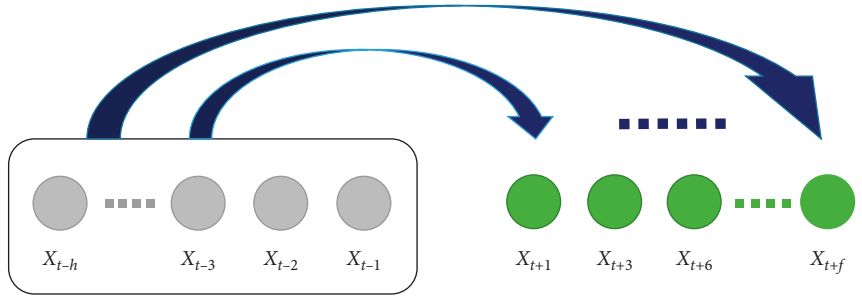

Figure 4: Multiscale predictor.

Meteorological (M) data features: <T, P0, P1, δP, H, WD, WP, dew>

Air pollutant (AP) data features: <FSP, NO_2, NO_x, O_3, RSP, SO_2>

Time data features: <month, week, day, hour>

The target is to forecast the near future AQHI. However, not all the features above are related to the AQHI, finding out the features which are correlated with the target would be beneficial. The historical pollutants and meteorology may impact the future air quality as the simple empirical approaches assume, finding out the influential historical time lag would be important as well.

(1) Feature Correlation Analysis. In this paper, Spearman's correlation analysis was used due to the possible nonlinear relationships between variables. Spearman's rank correlation coefficient measures the monotonic association between two variables and relies on the rank order of values [28]. The formula for Spearman's coefficient is

$$\rho_{\text{rank}_x,\text{rank}_y} = \frac{\text{cov}\left(\text{rank}_x, \text{rank}_y\right)}{\sigma_{\text{rank}_x}\sigma_{\text{rank}_y}}, \tag{16}$$

where $\text{rank}_x, \text{rank}_y$ are the ranked (sorted) values of variables x and y, $\text{cov}(\cdot)$ is the covariance, and $\sigma(\cdot)$ is the standard deviation. Figure 3 shows the Spearman correlation coefficients between the features of HK dataset. Correlation scores go from −1 to 1. Perfect positive correlation is 1. Perfect negative correlation is −1. The figure shows that FSP, O_3, RSP, SO_2, P0, and P1 have strong positive correlations with the AQHI, while T, H, and dew have strong negative correlations with the AQHI. Cohen's standard [29] was used in this paper to select the correlated features. Features with association smaller than 0.30 are discarded. The picked features are as follows:

<FSP, NO_2, NO_x, O_3, RSP, SO_2, T, P0, P1, δP, H, dew, WP, WD, month, hour>

(2) Temporal Correlation Analysis. Intuitively, historical data from different periods have different effects on future time lags. More recent events have a stronger influence on the current status, while earlier events have a weaker influence. Denote current time as t, the historical time lag as h, and the future time lag as f, and then the prediction time is $t + f$ ($f = 1, 3, 6, 9, 12$) and the influential historical time is $t - h$ ($h = 1, 2, \ldots, n$). The multiscale prediction task is represented in Figure 4. In this paper, the LSTM NN model which is capable of learning long time series was used to select the appropriate influential historical time lag [30].

The network architecture of the LSTM model used in the paper is shown in Figure 5, which is the same as the LSTM-extended network proposed in [13]. The main input is the air pollutant data, and the auxiliary input is the time and meteorology data. There are two LSTM layers and one

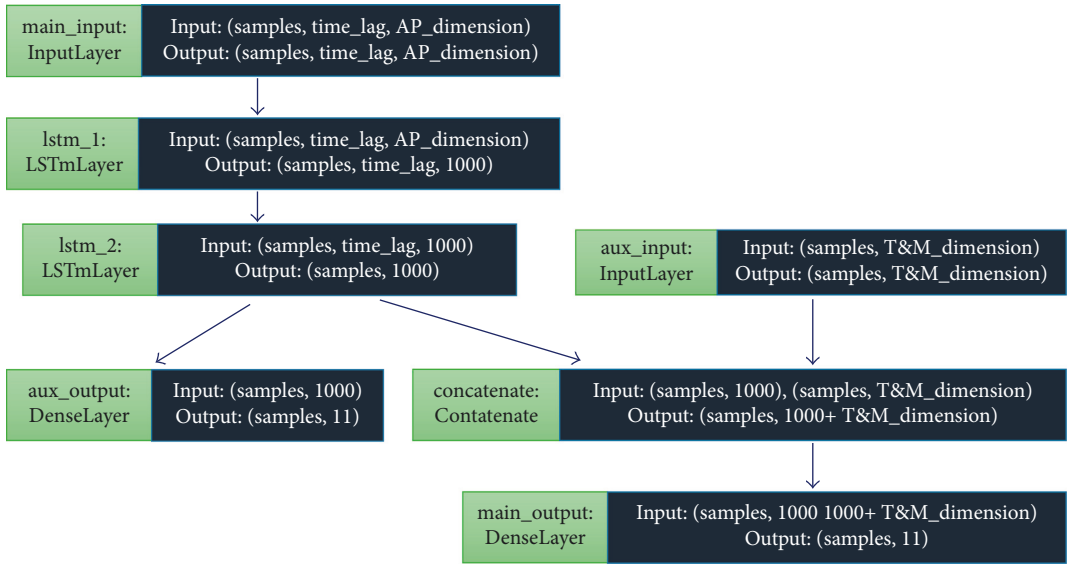

Figure 5: The LSTM network architecture used in this paper.

Table 4: Influences of different historical time lag over different future time lag.

f1	F-lag				
H-lag	1	3	6	9	12
0	0.659	0.543	0.452	0.409	0.388
1	0.647	0.488	0.483	0.446	0.408
2	0.603	0.573	0.443	0.417	0.395
3	0.606	0.528	0.467	0.434	0.431
4	0.642	0.465	0.439	0.452	0.417
5	0.706	0.654	0.511	0.454	0.413
6	0.739	0.504	0.574	0.486	0.423
7	0.711	0.700	0.682	0.508	0.436
8	0.736	0.705	0.713	0.447	0.481
9	**0.763**	0.721	**0.733**	**0.676**	0.512
10	0.674	**0.729**	0.693	0.557	0.495
11	0.630	0.702	0.632	0.631	0.541
12	0.728	0.656	0.584	0.607	**0.587**

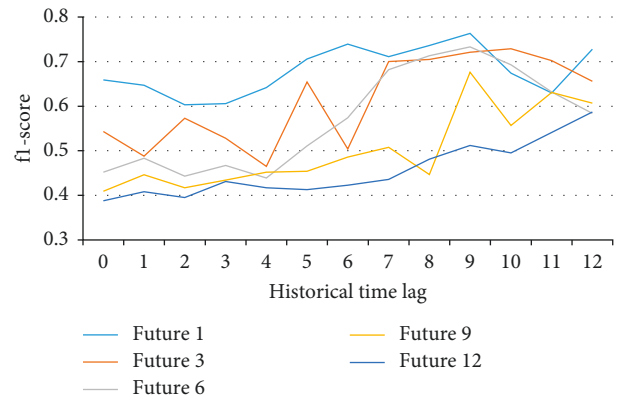

Figure 6: Influences of different historical time lag over different future time lag.

output layer which is a fully connected layer that has 11 neurons corresponding to the number of classes. The number of neurons in the LSTM layer has to be tuned. For simplicity, the number of neurons in each LSTM layer was set to an equivalent value chosen from a candidate set of {50, 100, 200, 500, 1000, 2000}. The most appropriate setting was chosen that yielded the best performance based on several comparative experiments. When the number of neurons in the LSTM was as 1000, the LSTM achieved the best performance. Therefore, in this paper, the number of neurons in the LSTM layers was set as 1000.

The future 1, 3, 6, 9, and 12 hours' AQHIs were predicted in this paper. With each future time lag, the influences of different historical time lags were examined. The results are given in Table 4. The evaluation metric is weighted f1-score (f1 in Table 4). The corresponding curve graph is given in Figure 6. The result shows that different future time lag (F-lag in Table 4) corresponds to slightly different optimal historical time lag (H-lag in Table 4). The general influential time of historical data for a specific future time's AQHI is around 9 hours.

Notably, the result shows that the prediction performances are poor for future time lag larger than 6, indicating that long-term prediction tasks are instinctively more difficult. Small-time lag cannot guarantee enough long-term memory inputs for the LSTM model, while large time lags permit an increased number of unrelated inputs, which increase the model's complexity and the difficulty of learning useful features. According to the above experiments, for simplicity, 9 was selected as the most appropriate influential historical time lag for different future time lag.

4. Results and Discussion

Algorithms used in the experiments are ARIMA, RF, MLP, SVC_linear (SVC with the liner kernel), SVC_rbf (SVC with the RBF kernel), SVC_sig (SVC with the sigmoid kernel), SVC_poly (SVC with the polynomial kernel), LSTM, and MKSVC. ARIMA was used as a baseline model, RF, MLP,

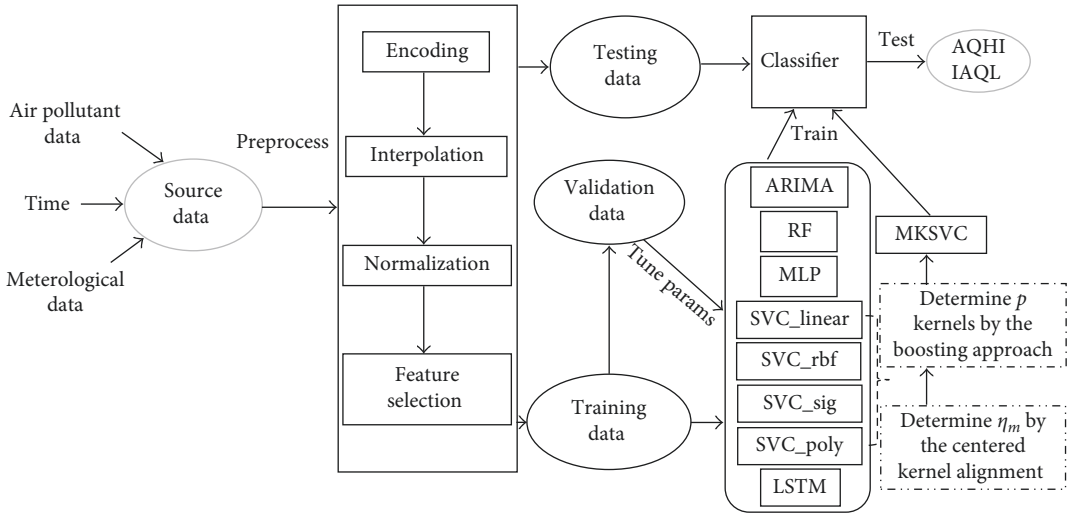

Figure 7: Experiment flow.

and SVC are widely used air quality forecast models, they were fine-tuned in this paper in order to make a fair comparison with MKSVC, and the LSTM in this paper has the same structure as the LSTM extended model proposed in [13]. Figure 7 shows the experimental flow. All algorithms were designed and tested with the same operation environment (Python 3.5.3, Windows 10, Intel® Core™ i7-5500U CPU @2.40 GHz, 16.0 GB RAM).

4.1. Parameter Optimization. Parameter optimization refers to the method of finding optimal parameters for a machine-learning algorithm. This is important since the performance of any machine learning algorithm depends to a huge extent on what the values of parameters are. For each prediction time lag, the parameters are different for each algorithm. It means an optimal model for each prediction task and each algorithm need to be tuned. The ways to get the parameters of MKSVC are detailed in Section 2 and Section 3.2.2 for LSTM. For the other algorithms, the parameter tuning process of the one-hour future time lag prediction task is presented in the following part, and the multiscale prediction tasks have identical fine-tuning processes.

First, the grid search interval of a parameter is narrowed by analyzing the influence curve of a single parameter on the training score and the validation score. For instance, by varying the kernel coefficient γ of the RBF kernel in SVC_rbf, the γ-score curve can be obtained as shown in Figure 8. The yellow line denotes the score over the training set. The purple line represents the score on the validation set, and the shadow represents the variance.

The figure shows that, at first, both the training and validation scores rise with the increase of γ. However, when γ reaches around 0.5, a further increase will result in the increase of the training score but the decrease of the validation score; it signifies that the model is getting overfitting. According to this influence curve, the grid search interval of γ in the next step can be narrowed between 0.0 and 1.0.

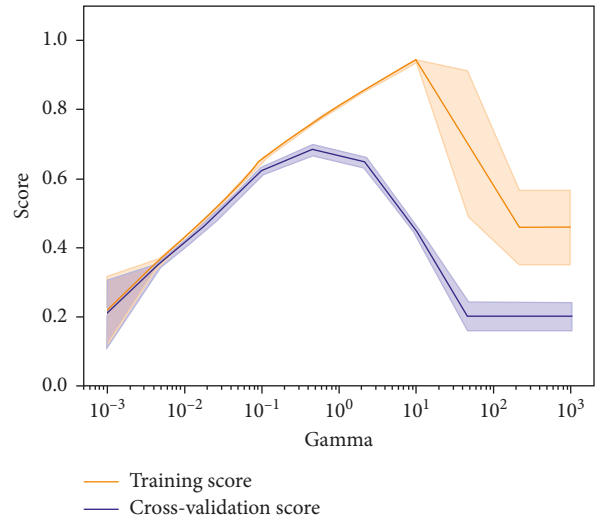

Figure 8: γ-score curve of SVC_rbf.

Based on the influence curve, the grid search intervals of the main parameters of the ARIMA, RF, MLP, and SVCs are shown in Table 5. RF, MLP, and SVCs used in this paper are implemented in scientific toolbox scikit-learn [31] and ARIMA implemented in statsmodels [32]. The unlisted parameters are set as default.

Then, a gird search with 5-fold cross validation was applied to find the optimum parameter. By exhaustively considering all parameter combinations in Table 5, the optimal parameter settings of the ARIMA, RFC, MLP, and SVCs are obtained as shown in Table 6. After getting the inner kernel coefficients of all the base kernels, the centered kernel alignment method described in Section 2.2. was used to get the optimal weight for each kernel.

4.2. Comparison. For HK, one year's data was used for training, and the other year's data was used for testing. For Beijing, the first two years' data was used for training, and

Table 5: Main parameters and their tuning range of the used algorithms.

Algorithm	Parameter	Algorithm	Parameter
ARIMA	p: [0,3], d: [0,10], q: [0,3]		
RF	n_estimators: [100, 1000; 50] max_depth: [10, 20; 1] max_features: [10, 30; 1] min_samples_split: [2,100; 1] min_samples_leaf: [1,100; 1]	SVC_linear	C: [100, 5000; 100]
		SVC_rbf	C: [100, 5000; 100] gamma: [0.0, 1.0; 0.01]
MLP	hidden_layer_sizes: {(50, 50), (100, 100), (10, 20, 10), (20, 40, 20)} activation: {'identity', 'logistic','tanh', 'relu'} solver: {'lbfgs', 'sgd', 'adam'}	SVC_sig	C: [100, 5000; 100] gamma: [0.0, 1.0; 0.01] coef0: [0, 1000; 50]
		SVC_poly	C: [100, 5000; 100] degree: {2,3} gamma: [0.0, 1.0; 0.01] coef0: [0, 1000; 50]

p: AR specification; d: integration order; q: MA specification; C: regularization coefficient in SVC; n_estimators: number of trees in the forest; max_depth: maximum depth of the tree; max_features: maximum number of features when looking for the best split; min_samples_split: the minimum number of samples required to split an internal node; min_samples_leaf: the minimum number of samples required to be at a leaf node; solver: algorithm used in the optimization problem; hidden_layer_sizes: hidden layer size; alpha: regularization term parameter in MLP; activation: activation function for the hidden layer; gamma: kernel coefficient for 'rbf', 'poly', and 'sigmoid'; degree: degree of the polynomial kernel function; coef0: independent term in kernel functions for 'poly' and 'sigmoid'; *[a, b; c] means within range [a, b], increase c every iteration; {} means set of values.

Table 6: The optimal parameter settings of the algorithms.

Algorithm	Parameter	Algorithm	Parameter
ARIMA	order (p,d,q): (2,0,2)		
RFC	n_estimators: 400 max_depth: 9 max_features: 11 min_samples_split: 95 min_samples_leaf: 71	SVC_linear	C: 400
		SVC_rbf	C: 300, gamma: 0.02
		SVC_sig	C: 100, gamma: 0.13, coef0: 400
MLP	hidden_layer_sizes: (20, 40, 20) activation: 'relu' solver: 'adam'	SVC_poly	C: 100, degree: 2, gamma: 0.04, coef0: 900
MKSVC	Kernel weights of linear, rbf, poly and sig kernels: (0.999, 0.212, 0,134, 0.00009)		

Table 7: Performance comparison for predicting the next hour's AQHI in HK.

	Accuracy	mse	wr	wf	wp
ARIMA	0.608	0.795	0.608	0.605	0.605
RF	0.782	0.279	0.782	0.779	0.782
MLP	0.908	0.101	0.908	0.908	0.909
SVC_linear	0.960	0.041	0.96	0.961	0.963
SVC_rbf	0.937	0.065	0.937	0.937	0.938
SVC_poly	0.959	0.042	0.959	0.959	0.961
SVC_sigmoid	0.267	4.996	0.267	0.113	0.071
LSTM	0.763	0.265	0.763	0.763	0.773
MKSVC	**0.972**	**0.030**	**0.972**	**0.971**	**0.972**

Table 8: Performance comparison for predicting the future 3 hour's AQHI in HK.

	Accuracy	Mse	wr	wf	wp
ARIMA	0.525	0.945	0.525	0.525	0.529
RF	0.782	0.275	0.782	0.778	0.781
MLP	0.938	0.09	0.938	0.936	0.935
SVC_linear	0.961	0.04	0.961	0.962	0.963
SVC_rbf	0.937	0.065	0.937	0.937	0.938
SVC_poly	0.954	0.047	0.954	0.955	0.956
SVC_sigmoid	0.267	4.994	0.267	0.113	0.071
LSTM	0.723	0.220	0.723	0.721	0.729
MKSVC	**0.974**	**0.028**	**0.974**	**0.975**	**0.974**

the other three year's data was used for testing. The comparisons of the predictions for the future 1, 3, 6, 9, and 12 hours are given below.

4.2.1. Predict the AQHI of Hong Kong. Tables 7–11 show the performances of the algorithms for forecasting the future 1, 3, 6, 9, and 12 hours' AQHI in Hong Kong. From the table, the following conclusions can be drawn:

(1) MKSVC performs best on all the three prediction tasks. SVC models with linear, RBF, and polynomial kernels perform better than other models except for the MKSVC. Sigmoid kernel SVC always makes the

Table 9: Performance comparison for predicting the future 6 hour's AQHI in HK.

	Accuracy	mse	wr	wf	wp
ARIMA	0.471	1.208	0.471	0.472	0.474
RF	0.785	0.27	0.785	0.781	0.783
MLP	0.942	0.086	0.942	0.939	0.938
SVC_linear	0.965	0.038	0.965	0.965	0.966
SVC_rbf	0.937	0.066	0.937	0.937	0.937
SVC_poly	0.959	0.043	0.959	0.960	0.960
SVC_sigmoid	0.267	4.992	0.267	0.113	0.071
LSTM	0.732	0.300	0.732	0.733	0.749
MKSVC	**0.976**	**0.028**	**0.976**	**0.976**	**0.976**

Table 10: Performance comparison for predicting the future 9 hour's AQHI in HK.

	Accuracy	mse	wr	wf	wp
ARIMA	0.467	2.020	0.467	0.433	0.436
RF	0.738	0.332	0.738	0.735	0.737
MLP	0.777	0.255	0.777	0.776	0.776
SVC_linear	0.799	0.231	0.799	0.798	0.799
SVC_rbf	0.787	0.244	0.787	0.786	0.786
SVC_poly	0.785	0.25	0.785	0.784	0.784
SVC_sigmoid	0.267	4.991	0.267	0.113	0.071
LSTM	0.681	0.393	0.692	0.676	0.645
MKSVC	**0.817**	**0.203**	**0.821**	**0.815**	**0.820**

Table 11: Performance comparison for predicting the future 12 hour's AQHI in HK.

	Accuracy	mse	wr	wf	wp
ARIMA	0.453	2.148	0.454	0.387	0.410
RF	0.559	0.806	0.559	0.554	0.551
MLP	0.597	0.66	0.597	0.598	0.603
SVC_linear	0.614	0.671	0.614	0.607	0.606
SVC_rbf	0.601	0.69	0.601	0.592	0.594
SVC_poly	0.59	0.716	0.590	0.585	0.583
SVC_sigmoid	0.267	4.992	0.267	0.113	0.071
LSTM	0.528	0.733	0.524	0.512	0.506
MKSVC	**0.630**	**0.609**	**0.641**	**0.629**	**0.633**

Table 12: Performance comparison for predicting the next hour's PM2.5 IAQL in Beijing.

	Accuracy	mse	wr	wf	wp
ARIMA	0.482	1.153	0.482	0.481	0.52
RF	0.472	1.956	0.472	0.442	0.443
MLP	0.486	1.686	0.486	0.466	0.465
SVC_linear	0.515	1.212	0.515	0.525	0.519
SVC_rbf	0.525	0.945	0.525	0.526	0.529
SVC_poly	0.520	1.033	0.520	0.520	0.521
SVC_sigmoid	0.391	3.999	0.391	0.219	0.153
LSTM	0.395	3.648	0.395	0.296	0.247
MKSVC	**0.605**	**0.806**	**0.605**	**0.605**	**0.620**

worst predictions which show that the sigmoid kernel is unable to capture the characters of the dataset.

(2) Time series models like ARIMA and LSTM fail to compete with the widely used parametric models like RF, MLP, and SVCs, and as the future time lag increases, the time series models' performances decrease, while the parametric models keep achieving very satisfying results.

(3) Among the well-performed SVC models, linear kernel model performs best, which demonstrates

Table 13: Performance comparison for predicting the future 3 hour's PM2.5 IAQL in Beijing.

	Accuracy	mse	wr	wf	wp
ARIMA	0.471	1.208	0.471	0.472	0.474
RF	0.477	1.858	0.477	0.454	0.451
MLP	0.491	1.678	0.491	0.482	0.477
SVC_linear	0.444	2.363	0.444	0.37	0.336
SVC_rbf	0.496	1.641	0.496	0.469	0.471
SVC_poly	0.489	1.760	0.489	0.462	0.464
SVC_sigmoid	0.391	3.999	0.391	0.219	0.153
LSTM	0.391	3.999	0.391	0.219	0.153
MKSVC	**0.525**	**0.945**	**0.525**	**0.525**	**0.529**

Table 14: Performance comparison for predicting the future 6 hour's PM2.5 IAQL in Beijing.

	Accuracy	mse	wr	wf	wp
ARIMA	0.442	2.381	0.442	0.367	0.332
RF	0.468	2.024	0.468	0.433	0.437
MLP	0.493	1.67	0.493	0.463	0.462
SVC_linear	0.451	2.207	0.451	0.385	0.408
SVC_rbf	0.500	1.595	0.500	0.477	0.478
SVC_poly	0.490	1.844	0.490	0.435	0.441
SVC_sigmoid	0.391	3.999	0.391	0.219	0.153
LSTM	0.397	3.701	0.396	0.253	0.167
MKSVC	**0.513**	**1.275**	**0.513**	**0.520**	**0.519**

Table 15: Performance comparison for predicting the future 9 hour's PM2.5 IAQL in Beijing.

	Accuracy	mse	wr	wf	wp
ARIMA	0.410	1.208	0.471	0.472	0.474
RF	0.457	1.909	0.457	0.446	0.439
MLP	0.48	1.706	0.48	0.457	0.456
SVC_linear	0.45	2.236	0.45	0.385	0.424
SVC_rbf	0.492	1.746	0.492	0.452	0.456
SVC_poly	0.482	1.813	0.482	0.453	0.45
SVC_sigmoid	0.39	4.000	0.39	0.219	0.152
LSTM	0.390	4.000	0.391	0.217	0.151
MKSVC	**0.507**	**1.133**	**0.510**	**0.505**	**0.500**

Table 16: Performance comparison for predicting the future 12 hour's PM2.5 IAQL in Beijing.

	Accuracy	mse	wr	wf	wp
ARIMA	0.386	4.290	0.391	0.355	0.318
RF	0.456	1.933	0.456	0.442	0.433
MLP	0.477	1.663	0.477	0.457	0.451
SVC_linear	0.451	2.204	0.451	0.386	0.400
SVC_rbf	0.489	1.858	0.489	0.431	0.425
SVC_poly	0.478	1.851	0.478	0.452	0.449
SVC_sigmoid	0.390	4.001	0.39	0.219	0.152
LSTM	0.383	4.360	0.387	0.202	0.147
MKSVC	**0.501**	**1.536**	**0.500**	**0.498**	**0.491**

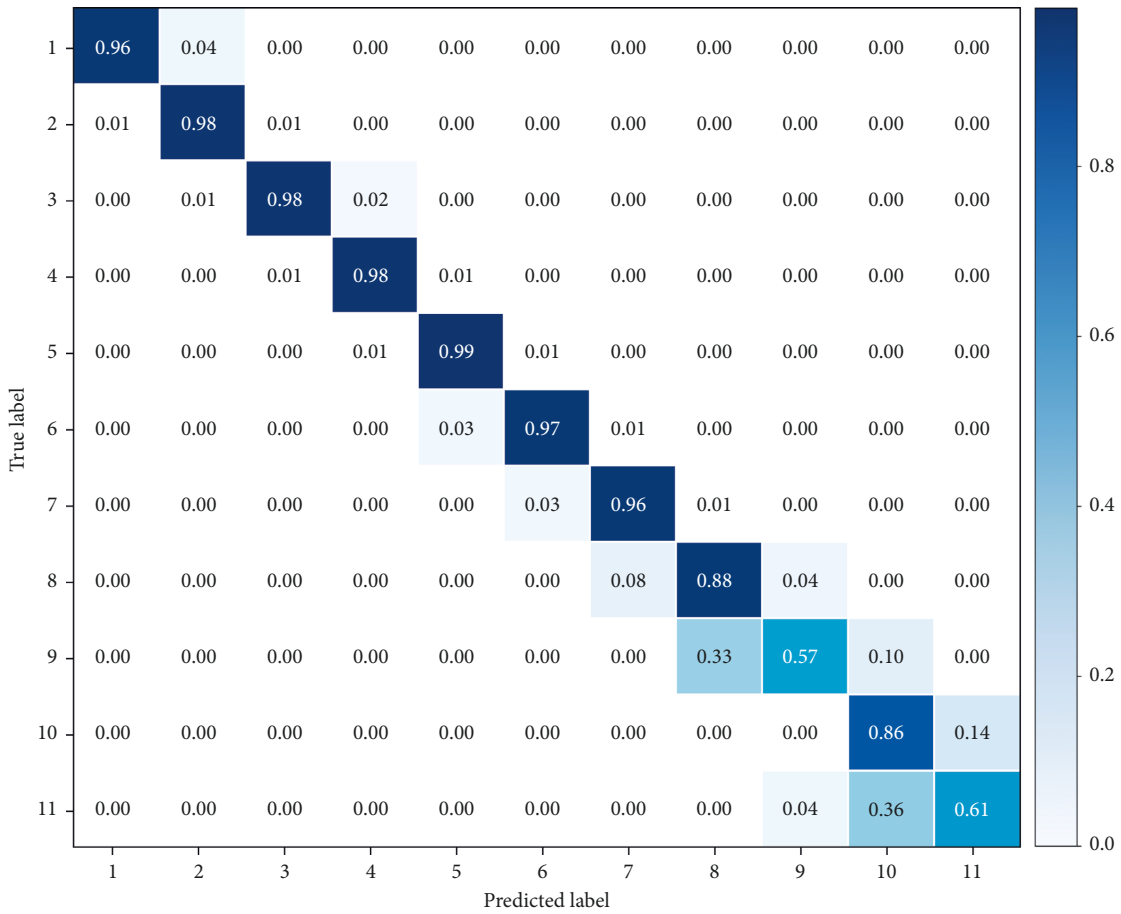

Figure 9: AQHI confusion matrix of MKSVC.

that the relation between the target and the input information has a lot of linear components, but there are also factors that influence the future air quality in a nonlinear way as the RBF and polynomial kernels also achieve promising performance.

(4) Models like MKSVC, MLP, and SVCs (except SVC_sigmoid) present very satisfying performance in the prediction for short-term air quality, larger than 90% of accuracy for the future 1, 3, and 6 hours. However, the performance for longer term predictions drops sharply from 0.976 of the 6 hour to 0.630 of the 12 hour (accuracy of MKSVC). It demonstrates that long-term air quality prediction is difficult.

4.2.2. Predict the PM2.5 IAQL of Beijing. Tables 12–16 show the performances of the algorithms for forecasting the next 1, 3, 6, 9, 12 hours' PM2.5 IAQL in Beijing. Similar conclusions can be drawn as that of HK, MKSVC is superior to other models, SVC_sigmoid and LSTM perform worst, SVCs behavior relatively better than other parametric models. But the overall performance of all the models on this dataset is much worse than that of HK. One possible reason is that there are fewer features in the Beijing dataset and the features in the dataset have a weaker correlation with the target. The other reason may be due to the generally worse air conditions in Beijing because higher polluting air conditions are harder to predict as demonstrated in the next part.

4.2.3. Comparison of Severe Air Pollution Prediction. Severe pollution prediction is a difficult task; however, it is critical as high-polluting air condition does way more damage to human health. Therefore, even a small improvement in the prediction of severe pollution is more meaningful than a large improvement in predicting good or less polluting air conditions.

As SVCs performed better than other algorithms except for the MKSVC, the best performing SVC was chosen to compare with the MKSVC in terms of forecasting severe air pollutions in the paper. AQHI greater than 6 is considered as severe pollution in HK. IAQL greater than 4 is considered as severe air pollution in Beijing. Figures 9 and 10 are the confusion matrixes of MKSVC and SVC_linear when predicting the next hour's AQHI of HK.

The x-axis denotes the predicted value, the y-axis denotes the true value, and the values on the diagonal of the matrix denote the probability of the correct prediction. The figures show that linear kernel SVC performs well in forecasting less polluting air conditions, so is the MKSVC. But MKSVC performs far better than linear kernel SVC when AQHI is larger than 8.

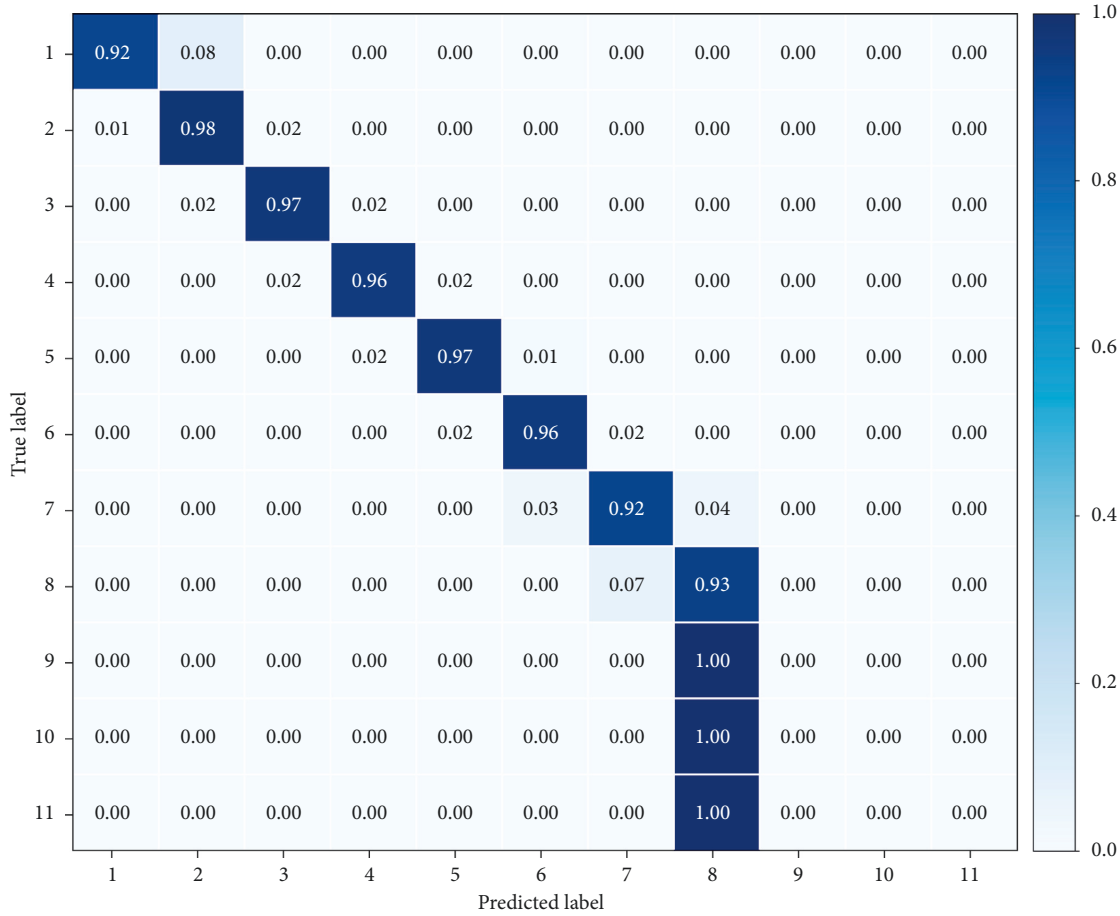

Figure 10: AQHI confusion matrix of SVC_linear.

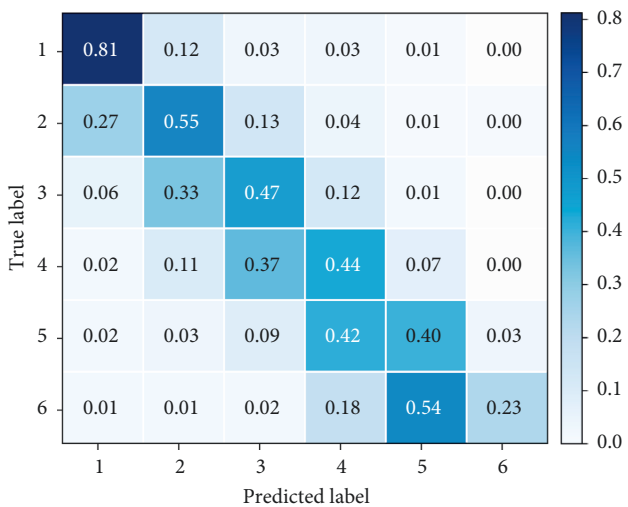

Figure 11: PM2.5 IAQL confusion matrix of MKSVC.

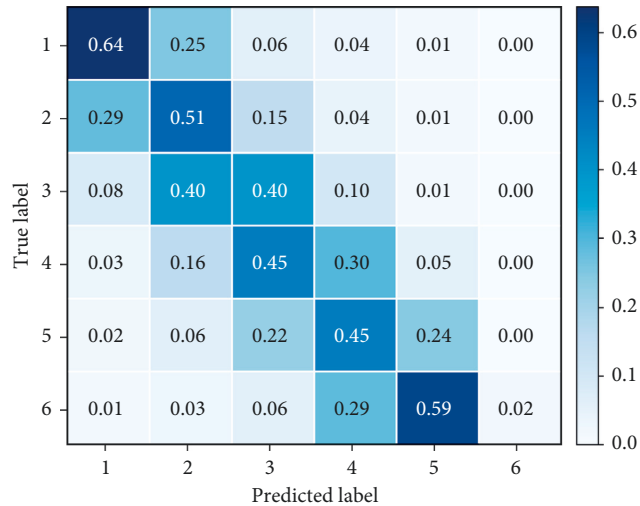

Figure 12: PM2.5 IAQL confusion matrix of SVC_rbf.

Figures 11 and 12 are the confusion matrixes of MKSVC and SVC_rbf when forecasting the next hour's PM2.5 IAQL of Beijing. The same conclusion can be drawn as that of HK. Generally, all the models make better prediction for light pollutions than severe ones due to the bias towards majority classes. It demonstrates that the task for severe air pollution prediction is challenging.

5. Conclusions

In this paper, a novel multiple kernel learning-based approach with SVC as the base learner was proposed for the near future's air quality prediction. It was the first time that multiple kernel learning method was

applied to air quality forecasting. Special attention was given to the feature engineering process. MKSVC is capable of learning the optimal combination of different kernels with which information coming from multiple sources can be captured simultaneously. Extensive experiments were conducted to compare the performance of MKSVC with the baseline model ARIMA, widely used parametric air quality forecasting models RFC, MLP, and SVCs, and a deep recurrent neural network model LSTM. Historical air pollutant concentration data, meteorological data, and time stamp data of a coastal city Hong Kong and an inland city Beijing were used to validate the models. Based on the experiments, a number of conclusions can be drawn:

(1) The proposed MKSVC algorithm offers a better predictive ability than the other models.

(2) The proposed MKSVC algorithm is capable of forecasting severe air pollution much better than the other models.

(3) The widely used parametric models RF, MLP, and SVC exhibit better prediction performance than the time series models ARIMA and LSTM.

(4) Feature transformation and feature selection play a significant role in making better air quality forecasting.

As can be seen from the experiments, long-term prediction task is difficult, so is the task to predict severe air pollutions. Though the proposed multiple kernel learning-based approach demonstrated relatively good performance in terms of both long-term prediction and severe air pollution prediction, more sophisticated methods need to be explored in order to build a more comprehensive and effective air quality forecasting system.

Appendix

A. Kernel SVM

Given a dataset with training instances $\{\mathbf{x}_i, y_i\}\,(i = 1, \ldots, N)$ where $\mathbf{x}_i$ is a vector in the input space $\mathbb{R}^D$ and y_i denotes the class index taking a value +1 or −1. SVM aims at minimizing an upper bound of the generalization error through maximizing the margin between the separating hyperplane and the data in the input space. In real-time problems, it is often not possible to determine an exact separating hyperplane dividing the data within the input space and also we might get a curved decision boundary in some cases. In such cases, the original input space can be mapped to a higher-dimensional feature space (Hilbert space) using nonlinear functions called feature functions ϕ: $\mathbb{R}^D \to \mathbb{R}^s$. The resulting discriminant function is

$$f(x) = \langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b. \tag{A.1}$$

The classifier can be trained by solving the following quadratic optimization problem:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}||\mathbf{w}||_2^2 + C\sum_{i=1}^{N}\xi_i, \\ \text{with respect to} \quad & \mathbf{w} \in \mathbb{R}^s,\ \xi \in \mathbb{R}_+^N,\ b \in \mathbb{R}, \\ \text{subject to} \quad & y_i\left(\mathbf{w}^T\phi(\mathbf{x}_i) + b\right) \geq 1 - \xi_i \quad \forall_i, \end{aligned} \tag{A.2}$$

where $\mathbf{w}$ is the vector of weight coefficients, b is the bias term of the separating hyperplane, C is a predefined positive trade-off parameter between model simplicity and classification error, ξ represents parameters for handling nonseparable data. Instead of solving this optimization problem directly, the Lagrangian dual function enables us to obtain the following dual formulation:

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \langle \phi(\mathbf{x}_i),\ \phi(\mathbf{x}_j)\rangle, \\ \text{with respect to} \quad & \alpha \in \mathbb{R}_+^N, \\ \text{subject to} \quad & \sum_{j=1}^{N}\alpha_i y_j = 0, \\ & C \geq \alpha_i \geq 0 \quad \forall_i, \end{aligned} \tag{A.3}$$

where α is the vector of dual variables corresponding to each separation constraint. Even though feature space is high dimensional, it could not be practically feasible to directly use the feature functions ϕ for classification of the hyperplane. So in such cases, nonlinear mapping induced by the feature functions is used for computation using special nonlinear functions called kernels.

$$\kappa(\mathbf{x}_i,\ \mathbf{x}_j) = \langle \phi(\mathbf{x}_i),\ \phi(\mathbf{x}_j)\rangle, \tag{A.4}$$

where κ: $\mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ is named the kernel function. By solving the above dual problem, we get $\mathbf{w} = \sum_{i=1}^{N}\alpha_i y_i \phi(\mathbf{x}_i)$, and the maximum margin separate hyperplane function can be rewritten as

$$f(\mathbf{x}) = \sum_{i=1}^{N}\alpha_i y_i \kappa(\mathbf{x}_i,\ \mathbf{x}) + b. \tag{A.5}$$

The multiclass support can be handled according to a one-versus-one or one-versus-rest scheme. The kernel trick allows SVMs to form nonlinear boundaries [14].

B. Calculation of AQHI in Hong Kong and IAQI in Mainland China

B.1. Calculation of AQHI. The AQHI of the current hour is calculated from the sum of the percentage added health risk (%AR) of daily hospital admissions attributable to the 3-hour moving average concentrations of four criteria air pollutants: ozone (O_3), nitrogen dioxide (NO_2), sulphur dioxide (SO_2), and particulate matter (PM) (respirable

Table 17: PM2.5 AQI of mainland China.

IAQI score	Description	PM2.5 concentration ($\mu g/m^3$)
0–50	Excellent	0–35
51–100	Good	35–75
101–150	Lightly polluted	75–115
151–200	Moderately polluted	115–150
201–300	Heavily polluted	150–250
301–500	Severely polluted	250–500

suspended particulates (RSP or PM10) or fine suspended particulates (FSP or PM2.5), whichever poses a higher health risk).

The %AR of each pollutant depends on its concentration and a risk factor which was derived from local health statistics and air pollution data. The %AR is then compared to a scale to obtain the appropriate banding of AQHI. The equations are as follows:

$$\begin{aligned}\%\mathrm{AR} = {} & \%\mathrm{AR}(\mathrm{NO_2}) + \%\mathrm{AR}(\mathrm{SO_2}) + \%\mathrm{AR}(\mathrm{O_3}) \\ & + \%\mathrm{AR}(\mathrm{PM}),\end{aligned} \tag{B.1}$$

where %AR (PM) = %AR (PM10) or %AR (PM2.5), whichever is higher.

$$\begin{aligned}\%\mathrm{AR}(\mathrm{NO_2}) &= \left[\exp\left(\beta(\mathrm{NO_2}) \times C(\mathrm{NO_2})\right) - 1\right] \times 100\%, \\ \%\mathrm{AR}(\mathrm{SO_2}) &= \left[\exp\left(\beta(\mathrm{SO_2}) \times C(\mathrm{SO_2})\right) - 1\right] \times 100\%, \\ \%\mathrm{AR}(\mathrm{O_3}) &= \left[\exp\left(\beta(\mathrm{O_3}) \times C(\mathrm{O_3})\right) - 1\right] \times 100\%, \\ \%\mathrm{AR}(\mathrm{PM10}) &= \left[\exp\left(\beta(\mathrm{PM10}) \times C(\mathrm{PM10})\right) - 1\right] \times 100\%, \\ \%\mathrm{AR}(\mathrm{PM2.5}) &= \left[\exp\left(\beta(\mathrm{PM2.5}) \times C(\mathrm{PM2.5})\right) - 1\right] \times 100\%,\end{aligned} \tag{B.2}$$

where %AR (NO_2), %AR (SO_2), %AR (O_3), %AR (PM), %AR (PM10), and %AR (PM2.5) are the added health risk of NO_2, SO_2, O_3, PM, PM10, and PM2.5 respectively;

C(NO_2), C(SO_2), C(O_3), C(PM10), and C(PM2.5) are the 3-hour moving average concentration of the respective pollutants in microgram per cubic meter ($\mu g/m^3$). $\beta(NO_2) = 0.0004462559$, $\beta(SO_2) = 0.0001393235$, $\beta(O_3) = 0.0005116328$, $\beta(\mathrm{PM10}) = 0.0002821751$, and $\beta(\mathrm{PM2.5}) = 0.0002180567$ are added health risk factors (technically known as regression coefficients) of the respective pollutants [24].

B.2. Calculation of IAQI. Each pollutant's individual AQI is called its IAQI. The highest IAQI among these six pollutants at a given time is called the primary or dominant pollutant and is chosen for the overall AQI value.

$$\begin{aligned}\mathrm{IAQI}_p &= \frac{\mathrm{IAQI}_{H_i} - \mathrm{IAQI}_{L_o}}{\mathrm{BP}_{H_i} - \mathrm{BP}_{L_o}}\left(C_p - \mathrm{BP}_{L_o}\right) + \mathrm{IAQI}_{L_o}, \\ \mathrm{AQI} &= \max\{\mathrm{IAQI}_1,\ \mathrm{IAQI}_2,\ \mathrm{IAQI}_3,\ \ldots,\ \mathrm{IAQI}_n\},\end{aligned} \tag{B.3}$$

where C_p is mass concentration value of the air pollutant p, BP_{H_i} is the high value of the concentration limit which can be checked in the reference table from the paper [25], BP_{L_o} is the low value of the concentration limit which can be checked in the reference table from [25], IAQI_{H_i} is the corresponding value of BP_{H_i} in the same reference table, and IAQI_{L_o} is also the corresponding value of BP_{L_o} in the reference table. The detailed break down of China AQI for PM2.5 concentrations is shown in Table 17.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

Hong Zheng is the group leader and she is responsible for the project management and in charge of revising this manuscript. Haibin Li is responsible for data analysis and planning and performing the experiments. Xingjian Lu and Tong Ruan provided valuable advice about the revised manuscript.

Acknowledgments

The authors are pleased to acknowledge the National Natural Science Foundation of China under Grant nos. 61103115 and 61103172; the National Natural Science Youth Foundation of China under Grant no. 61602175; the special fund for Software and Integrated Circuit Industry Development of Shanghai under Grant no. 150809; and the "Action Plan for Innovation on Science and Technology" Projects of Shanghai (Project no. 16511101000).

References

[1] Y. Wang, Y. Han, T. Zhu, W. Li, and H. Zhang, "A prospective study (SCOPE) comparing the cardiometabolic and respiratory effects of air pollution exposure on healthy and prediabetic individuals," *Science China Life Sciences*, vol. 60, no. 1, pp. 46–56, 2017.

[2] G. Cohen, I. Levy, Yuval et al., "Long-term exposure to traffic-related air pollution and cancer among survivors of myocardial infarction: a 20-year follow-up study," *European Journal of Preventive Cardiology*, vol. 24, no. 1, pp. 92–102, 2017.

[3] T. S. Dye, *Guidelines for Developing an Air Quality (Ozone and PM2. 5) Forecasting Program*, vol. 4, pp. 206-207, United States Environmental Protection Agency, Washington, DC, USA, 2013.

[4] Y. Zhang, M. Bocquet, V. Mallet, C. Seigneur, and A. Baklanov, "Real-time air quality forecasting, part I: history, techniques, and current status," *Atmospheric Environment*, vol. 60, pp. 632–655, 2012.

[5] U. Kumar and V. K. Jain, "ARIMA forecasting of ambient air pollutants (O_3, NO, NO_2 and CO)," *Stochastic Environmental Research and Risk Assessment*, vol. 24, no. 5, pp. 751–760, 2010.

[6] A. Saxena and S. Shekhawat, "Ambient air quality classification by grey wolf optimizer based support vector machine," *Journal of Environmental and Public Health*, vol. 2017, Article ID 3131083, 12 pages, 2017.

[7] C. M. Vong, W. F. Ip, P. K. Wong, and J. Y. Yang, "Short-term prediction of air pollution in Macau using support vector machines," *Journal of Control Science and Engineering*, vol. 2012, Article ID 518032, 11 pages, 2012.

[8] X. Hu, J. H. Belle, X. Meng et al., "Estimating PM2.5 concentrations in the conterminous United States using the random forest approach," *Environmental Science & Technology*, vol. 51, no. 12, pp. 6936–6944, 2017.

[9] R. Yu, Y. Yang, L. Yang, G. Han, and O. A. Move, "RAQ—a random forest approach for predicting air quality in urban sensing systems," *Sensors*, vol. 16, no. 1, p. 86, 2016.

[10] A. Russo, P. G. Lind, F. Raischel, R. Trigo, and M. Mendes, "Neural network forecast of daily pollution concentration using optimal meteorological data at synoptic and local scales," *Atmospheric Pollution Research*, vol. 6, no. 3, pp. 540–549, 2015.

[11] K. Karatzas, N. Katsifarakis, C. Orlowski, and A. Sarzyński, "Urban air quality forecasting: a regression and a classification approach," in *Proceedings of the Asian Conference on Intelligent Information and Database Systems*, vol. 2017, pp. 539–548, Springer, Kanazawa, Japan, April 2017.

[12] E. Pardo and N. Malpica, "Air quality forecasting in Madrid using long short-term memory networks," in *Proceedings of the International Work-Conference on the Interplay between Natural and Artificial Computation*, vol. 2017, pp. 232–239, Springer, Corunna, Spain, June 2017.

[13] X. Li, L. Peng, X. Yao et al., "Long short-term memory neural network for air pollutant concentration predictions: method development and evaluation," *Environmental Pollution*, vol. 231, pp. 997–1004, 2017.

[14] V. Vapnik, "The support vector method of function estimation," in *Nonlinear Modeling: Advanced Black-Box Techniques*, J. A. K. Suykens and J. P. L. Vandewalle, Eds., vol. 55, p. 86, Springer, New York City, NY, USA, 1998.

[15] C. Cortes, M. Mohri, and A. Rostamizadeh, "Algorithms for learning kernels based on centered alignment," *Journal of Machine Learning Research*, vol. 13, pp. 795–828, 2012.

[16] F. Aiolli and M. Donini, "EasyMKL: a scalable multiple kernel learning algorithm," *Neurocomputing*, vol. 169, pp. 215–224, 2015.

[17] S. Niazmardi, B. Demir, L. Bruzzone, A. Safari, and S. Homayouni, "Multiple kernel learning for remote sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 3, pp. 1425–1443, 2017.

[18] Y. Zhang, H.L. Yang, S. Prasad, E. Pasolli, J. Jung, and M. Crawford, "Ensemble multiple kernel active learning for classification of multisource remote sensing data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 2, pp. 845–858, 2015.

[19] H. Wen, Y. Liu, I. Rekik et al., "Multi-modal multiple kernel learning for accurate identification of Tourette syndrome children," *Pattern Recognition*, vol. 63, pp. 601–611, 2017.

[20] M. Gönen and E Alpaydın, "Multiple kernel learning algorithms," *Journal of Machine Learning Research*, vol. 12, pp. 2211–2268, 2011.

[21] Aqhi.gov.hk, Environmental Protection Department, July 2017, http://epic.epd.gov.hk.

[22] RP5.ru: Weather for 243 Countries of the World, July 2017, http://rp5.ru.

[23] Uci.edu: Machine Learning Repository, Beijing PM2.5 Dataset, https://archive.ics.uci.edu.

[24] W. T. Wai, W. T. W. San, M. A. W. H. Shun et al., "A study of the air pollution index reporting system," *Statistical Modelling*, vol. 13, p. 15, 2012.

[25] F. Gao, "Evaluation of the Chinese new air quality index (GB3095-2012): based on comparison with the US AQI system and the WHO AQGs," Bachelor's thesis, Integrated Coastal Zone Management, Raseborg, Finland, 2013.

[26] Q. W. Yan, "Environmental protection department issued HJ633–2012 environmental air quality index (AQI) technical requirements (trial)," *CSG*, vol. 4, p. 49, 2012.

[27] D. M. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.

[28] J. H. McDonald, *Handbook of Biological Statistics*, Sparky House Publishing, Baltimore, MD, USA, 2009.

[29] J. Cohen, P. Cohen, S. G. West et al., *Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences*, Routledge, Abingdon, UK, 2013.

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

[31] F. Pedregosa, G. Varoquaux, A. Gramfort et al., "Scikit-learn: machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[32] S. Seabold and P. Josef, "Statsmodels: econometric and statistical modeling with Python," in *Proceedings of the 9th Python in Science Conference*, Austin, TX, USA, June 2010.