# Stacked Selective Ensemble for $PM_{2.5}$ Forecast

Ke Gu, Zhifang Xia, and Junfei Qiao, *Member, IEEE*

***Abstract*— In this paper, we propose a new stacked selective ensemble-backed predictor (SSEP) to forecast the concentration of $PM_{2.5}$ based on the impact of measurements of the known air pollutants and meteorological data on the unknown $PM_{2.5}$ concentration over the following 48 h. It was found that a single learner cannot validly uncover and model the relationship between the future $PM_{2.5}$ concentration and the current and historical meteorological and pollutant data, mainly because any individual learner has limitations, especially facing to highly complex and ever-changing environmental problems, such as $PM_{2.5}$ prediction. Ensemble methods, which are widely acknowledged to yield strong generalization ability by boosting weak learners, are used in this paper to solve the aforesaid challenge. Our solution, aligned with an analysis of influencing factors on the future $PM_{2.5}$ concentration, generates multiple component learners for aggregation by introducing three types of diversities. Then, we adopt a pruning method to remove the negative component learners in each diverse type according to dynamic thresholds, which are determined by jointly considering the performance of each individual learner and the correlations between each pair of learners. A stacking technique is finally applied to the selected positive component learners to forecast the $PM_{2.5}$ concentration in the future. Thorough experiments demonstrate the superiority of our proposed SSEP in contrast to relevant state-of-the-art models when applied to $PM_{2.5}$ prediction.**

***Index Terms*— Air pollutants, diversity, fine particulate matter ($PM_{2.5}$), meteorological factors, selective ensemble, stacking.**

## I. Introduction

DURING recent decades, high-speed industrialization has massively facilitated people but simultaneously introduced many passive influences, such as environmental pollution, resource shortage, and ecological damage. Among these influences, environmental pollution caused by man- or nature-made destructive behavior forces exceedingly hazardous substances into our living environment, surpassing the environment's self-purification ability. Air, soil, and water pollutions are the three typical pollution problems. Compared with the latter two, polluted air is very likely to give rise to a greater risk not only deteriorating human health but also contributing to panic within all of society, primarily due to its ubiquity. Removing air pollutants and improving air quality might be incapable of being fulfilled in the short term but may be a chronic project. Therefore, one of the dominant tasks currently is to rely on air quality prediction that can guide people toward healthy travel and facilitate the decision-making of governments toward production halts or traffic restrictions, and so forth. Hence, an efficient and effective predictor of air pollutant concentration is strongly desired.

Fig. 1. Examples of typical $PM_{2.5}$ monitors.

The main ways of polluting air include some anthropogenic-caused harmful gases and particulates emitted due to motor vehicles, the processes of steel making, oil refining, and pharmaceuticals, as well as the combustion of oil, coal, and natural gas [1]. Six of the most frequently seen air contaminants consist of $NO_2$, $SO_2$, $O_3$, CO, $PM_{2.5}$ [fine particulate matter (PM)], and $PM_{10}$ (respirable PM). The first four gaseous pollutants readily result in respiratory inflammation and nervous system disorder when the concentration exceeds a certain degree. In contrast, the remaining two types of PM refer to the particles with the aerodynamic diameters smaller than or equivalent to 2.5 and 10 $\mu$m. It is evident that $PM_{2.5}$ is the component of $PM_{10}$, and its index is less than $PM_{10}$ [2]. In comparison with respirable PMs, fine PM easily intrudes into human lungs and is cleansed with difficulty. With constant exposure in environments with high-concentration $PM_{2.5}$, the morbidity and mortality of the public inevitably grow. Thus, more research attention has been paid to the monitoring of $PM_{2.5}$ [3], [4] and relevant instruments [5], [6], as shown in Fig. 1.

Manuscript received August 25, 2018; revised January 19, 2019; accepted February 27, 2019. This work was supported in part by the National Science Foundation of China under Grant 61703009 and 61890930-5, in part by the China Association for Science and Technology through the Young Elite Scientist Sponsorship Program under Grant 2017QNRC001, in part by the Beijing Excellent Talents Funding through the Young Top-Notch Talents Team Program under Grant 2017000026833ZK40, in part by Major Science and Technology Program for Water Pollution Control and Treatment of China under Grant 2018ZX07111005, and in part by National Key Research and Development Project under Grant 2018YFC1900800-5. The Associate Editor coordinating the review process was Huang-Chen Lee. *(Corresponding author: Ke Gu.)*

K. Gu and J. Qiao are with the Beijing Advanced Innovation Center for Future Internet Technology, Beijing Key Laboratory of Computational Intelligence and Intelligent System, Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China (e-mail: guke.doctor@gmail.com; junfeiq@bjut.edu.cn).

Z. Xia is with the Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China, and also with the State Information Center of China, Beijing, China (e-mail: spidergirl21@163.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIM.2019.2905904

The variations of $PM_{2.5}$ concentration follow a complicated process. Many studies are being devoted to unveiling the composition of influencing meteorological factors and pollutants, as well as the mechanism of impact between the above-mentioned meteorological factors and pollutants at present and in historical times and the $PM_{2.5}$ concentration that occurs later. A great number of good works were proposed based on computational dynamic models to predict pollutants [7], [14]. Lu *et al.* [11], [12] provided a 3-D system for modeling the urban and regional air pollution from the following four main aspects: a meteorological model, a tracer transport code, a chemical and aerosol microphysical model, and a radiative transfer code. Huang and Tai [13] and Wang and Ogawa [14] discussed several typical meteorological parameters, e.g., temperature, wind speed, wind direction, humidity, and pressure, and their impacts on the variations of $PM_{2.5}$ concentration.

However, a growing number of studies have been devoted to data-driven air quality forecasts since computational dynamic models are neither easily fitted nor suitable for highly developed and overpopulated cities such as Beijing. In contrast with computational dynamic models, data-driven methods are better at mining the underlying relationships and thus have aroused extensive attention during recent years. Ordieres *et al.* [15] and Kumar and Jain [16] analyzed the correlations of typical air pollutants in the temporal domain and forecasted the changing tendency of air pollutant indices in the future.

Despite the competitive results attained by the aforementioned works, limited efforts have been made to validly incorporate environmental and temporal features that were found to be closely correlated with the future $PM_{2.5}$ concentrations [17]. Elbayoumi *et al.* [18] applied multiple linear regression, principal component analysis, and principal component regression to infer the indoor $PM_{10}$ and $PM_{2.5}$ concentration on the grounds of relevance in environmental and temporal features. However, such linear models pose difficulties when employed to describe the complicated and ever-changing environmental problems, such as air quality prediction [19], and thus, nonlinear models with substantial descriptive powers are expected to address these problems. In [20]–[22], single machine learners, such as neural networks, were introduced to well approximate the relationship between the future $PM_{2.5}$ concentration and the current and historical records of meteorological and pollutant parameters.

However, some major limitations in the above-mentioned studies exist. First, those predictive models only use linear models or simply concatenate linear and nonlinear models, which scarcely address such a highly complex problem as an air quality forecast. Second, those predictive models solely take advantage of temporal features or environmental features or indiscriminately combine temporal features and environmental features as equivalent inputs into the regressors. Aiming at resolving these problems, we adopted an application (App) to gather a considerable number of meteorological and pollutant data. In terms of exploration and analysis of our collected samples, the influencing factors on the future $PM_{2.5}$ concentration can be roughly categorized into three respects, i.e., environmental factors (e.g., windy or not), temporal factors (e.g., $PM_{2.5}$ values during the past hours), and selected samples. Regarding the above-mentioned concerns, this paper proposes a novel stacked selective ensemble-backed predictor (SSEP) of the $PM_{2.5}$ concentration in the future. More specifically, the proposed predictor is implemented in a three-stage framework. First, we create several component learners with popular methods (e.g., support vector regressor (SVR) [23]) to address the three categories of diversities, which are produced by appropriately selecting environmental factors, temporal factors, and training samples. Second, we employ a pruning technique to delete negative component learners in each of the three categories in accordance with dynamic thresholds, which are determined by jointly taking the performance of each individual learner as well as the correlations between each pair of learners into consideration. Third, we apply a stacking method on the selected positive component learners for aggregation and thus predict the $PM_{2.5}$ concentration.

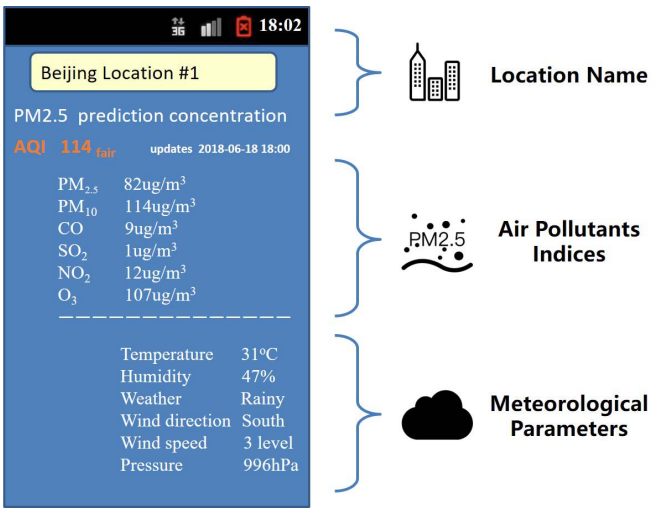



Fig. 2. App for gathering records of meteorological and pollutant historical data.

The layout of this paper is outlined as follows. Section II first presents how to collect measurements of meteorological variables and air pollutants and then describes the proposed SSEP predictor of the future $PM_{2.5}$ concentrations in detail. In Section III, we conduct the performance comparison between our SSEP model and state-of-the-art predictive models using the collected data samples. Section IV describes the conclusions and discusses future work.

## II. Proposed Predictive Model

Recent years have witnessed the rapid development of industrialization in many countries, accompanied by boosting environmental problems such as $PM_{2.5}$, which highly concern governments and people. Severe pollution does serious harm to people's health and safety. Therefore, a reliable predictor of future $PM_{2.5}$ concentrations is strongly required. This section will first introduce how to collect data and the corresponding analysis, followed by providing the proposed stacked selective ensemble-based SSEP predictor of future $PM_{2.5}$ concentrations.

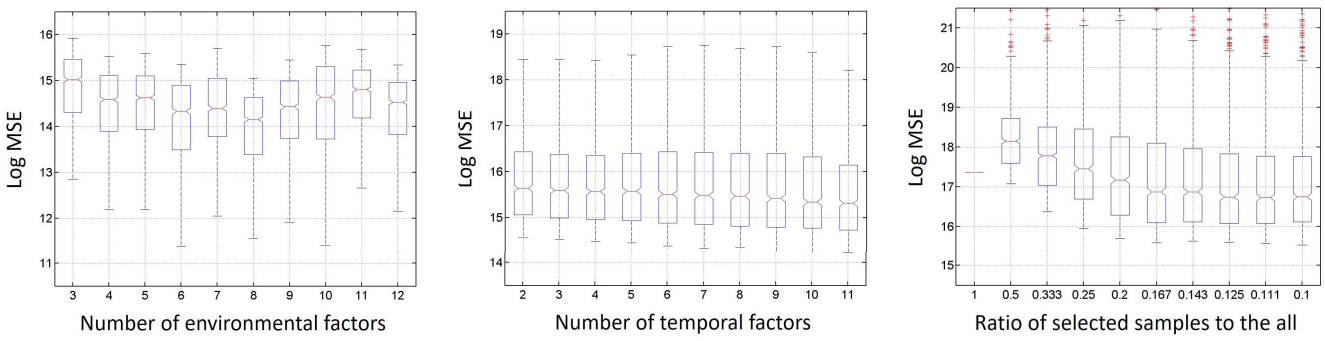



Fig. 3. Box plot of the distribution of log MSE across (a) different numbers of environmental factors, (b) different numbers of temporal factors, and (c) different ratios of the selected samples to the whole data samples.

### *A. Data Collection and Analysis*

We focus on the variations in $PM_{2.5}$ concentration in 12 typical locations in the city of Beijing, the capital of China. We collected the hourly measurements of six air contaminants (including $NO_2$, $O_3$, $PM_{2.5}$, $PM_{10}$, CO, and $SO_2$) and six meteorological variables (including humidity, temperature, pressure, weather, wind speed, and wind direction) during the last year. Tens of thousands of groups of measurements were collected and deployed to establish and examine the predictive model of $PM_{2.5}$ concentration. An elaborately designed Android App using JAVA language was applied to access to [24], [25] each hour to automatically collect the hourly records of meteorological indices and air contaminants, as exhibited in Fig. 2. The hourly records are the average values within 1 h. A portion of the indicators with their associated units is as follows: $PM_{2.5}$ ($\mu g/m^3$), $PM_{10}$ ($\mu g/m^3$), CO ($\mu g/m^3$), $SO_2$ ($\mu g/m^3$), $NO_2$ ($\mu g/m^3$), $O_3$ ($\mu g/m^3$), temperature (°C), humidity (%), and pressure (hPa). There are 16 types of weather, and their values and the associated physical meanings are 0 (sunny), 1 (cloudy), 2 (overcast), 3 (rainy), 4 (sprinkle), 5 (moderate rain), 6 (heavy rain), 7 (rain storm), 8 (thunder storm), 9 (freezing rain), 10 (snowy), 11 (light snow), 12 (moderate snow), 13 (heavy snow), 14 (foggy), 15 (sand storm), and 16 (dusty). There are eight types of wind direction, and their values and the associated physical meanings are 0 (north wind), 1 (northwest wind), 2 (west wind), 3 (southwest wind), 4 (south wind), 5 (southeast wind), 6 (east wind), and 7 (northeast wind). The wind speed includes 18 levels, from 0 (no wind) to 17 (super typhoon). Note that aerosol optical thickness is not a typical meteorological index but is measured and retrieved using LiDAR [26], and thus, it is not included in this paper. The concentrations of $NO_2$, $SO_2$, $O_3$, CO, $PM_{2.5}$, and $PM_{10}$ were separately recorded by using the TE-42CTL NO-$NO_2$-$NO_x$ analyzer, TE43C $SO_2$ analyzer, TE-49C $O_3$ analyzer, TE48C CO analyzer, and TEOM P1400a, respectively.

The majority of traditional predictive models of $PM_{2.5}$ concentration does not pay attention to the specific characteristics of meteorological and pollutant data but merely resort to several complicated learning tools or their hybrid models. Neural networks can be considered an example. Evidently, shallow neural networks are unable to build a good mapping from the meteorological and pollutant historical data to the future $PM_{2.5}$ concentration, i.e., they result in underfitting or large bias on the training data set and are similarly problematic on the testing data set. In contrast, complicated deep neural networks, because of their use of deep layers and massive neurons, have a substantial descriptive ability and can better predict the $PM_{2.5}$ concentration by learning the meteorological and pollutant historical data on the training set; however, they might introduce overfitting or a large variance on the testing set and thereby have inferior generalization ability. One good solution to combat the problem illustrated above is to fuse complicated learning models with big data. Zheng *et al.* [27] proposed an excellent approach for forecasting air quality based on a four-component model, involving a temporal predictor, a spatial predictor, a dynamic aggregator, and an inflection predictor. Very recently, Soh *et al.* [28] provided an extraordinary deep learning-based work for air quality prediction, which merges the popular neural network architectures of a convolutional neural network and a long short-term memory (LSTM) model and results in very impressive prediction performance. Motivated by these two studies, the other solution is to deeply mine the characteristics of meteorological and pollutant data, and on this basis, multiple specific simple learning models are devised to be fused to derive an overall air quality prediction system. In this work, we adopt the second solution and will first make comprehensive analyses on meteorological and pollutant data in what follows.

Based on the measured data collected by our App, we have derived the subsequent three observations. First, we find that using the entire set of environmental factors to predict $PM_{2.5}$ is not always superior to using a portion of them. For a vivid illustration, we randomly select 1000 sample sets, each of which contains at least 200 independent data samples. For each sample set, we randomly choose $k$ (e.g., $k = 3$) environmental features from all the environmental factors to forecast the $PM_{2.5}$ concentration 24 h later. The number $k$ is assigned in the ordered sequence $3, 4, \ldots, 12$. Based on the popular SVR provided in the LibSVM package [29], 60% of one sample set is employed for training, 20% percent for validation, and the remaining 20% for testing. The prediction accuracy is computed using the mean squared error (MSE). An elegant predictive model is expected to obtain an MSE value close to 0. For a certain value of $k$, we obtain 1000 MSE values and show the associated box plot of the distribution of log MSE in Fig. 3(a). Similarly, the log MSE values of the other

nine values of $k$ can be derived, and their box plots are presented in Fig. 3(a) for comparison. One can see that in some scenarios, using a portion of the proper features (e.g., $k = 8$) can generate greater performance and higher stability than directly using the entire 12 features. Furthermore, we also find that in the feature sets with the top 5% performance, $NO_2$ has the lowest occurrence frequency, much smaller than that of the other features.

The second observation concerns predicting the $PM_{2.5}$ concentration 24 h later, in which case the use of more historical $PM_{2.5}$ data is able to deliver better prediction accuracy. Equivalently, we use the same 1000 sample sets as used in the first context. For each sample set, we separately select $l$ temporal features, where $l = 2, 3, \ldots, 11$ is the number of features, which means that we apply the $PM_{2.5}$ data at the current moment and in the previous $l - 1$ hours to forecast the $PM_{2.5}$ concentration 24 h later. For instance, when $l$ is assigned as 3, we use three features, including the $PM_{2.5}$ data at the current moment $T_0$ and the previous moments $T_{-1}$ and $T_{-2}$ to predict the $PM_{2.5}$ concentration at $T_{24}$. The same setting is adopted for training, validation, and testing, as well as the evaluation index for computing the prediction accuracy. For each of $l$ value, we acquire 1000 MSE values and show the associated box plot of the distribution of log MSE in Fig. 3(b). As shown in the figure, using a larger quantity of historical $PM_{2.5}$ data can lead to greater prediction accuracy, but the gain tends to be small as $l$ grows.

The last consideration is the influence of selected samples on the performance of the $PM_{2.5}$ predictor. We randomly choose 1000 sample sets, each of which contains $1/m$ of all the data samples, where $m$ is set as $1, 2, \ldots, 10$. We use the 12 environmental features measured at the current time, the SVR for learning, and the MSE for evaluation, and set the ratio of training, validation, and testing sets to 3:1:1, the same as used earlier. For each of the $m$ values (expect $m = 1$), 1000 MSE values are computed, and the associated box plot of the distribution of log MSE is shown in Fig. 3(c). We observe that in some cases, using a portion of the samples for the $PM_{2.5}$ forecast is superior to applying the overall data samples.

### *B. Stacked Selective Ensemble for $PM_{2.5}$ Forecast*

According to the analyses stated above, appropriately using selected samples, environmental factors, and temporal factors is beneficial to devising a well-designed predictor of $PM_{2.5}$ concentration. The ensemble learning is good at solving this problem, and it will become better if an appropriate pruning technique is used before aggregation [30]. Basically, two of the most typically used ensemble learning methods are bootstrap aggregating (bagging) [31] and random subspace [32], which have been widely applied in industrial Apps [33]–[35], image processing [36]–[38], remote sensing [39], [40], and so on. In fact, these two methods are independently suitable to be applied to selected samples and environmental factors to forecast future $PM_{2.5}$ concentrations. In what follows, the proposed SSEP model will be illustrated from the subsequent three steps.

First, we generate component learners based on the three categories of diversities that originate from selected samples, environmental factors, and temporal factors. We apply bagging, which was proposed to combine the benefits of bootstrapping and aggregation [31], to the selected samples. Bootstrapping is used to create multiple sets of samples by randomly sampling with the replacement of the original training data, and then, multiple learners are generated by training on the above multiple sets. In general, each of the multiple sets has the same size as the original training data. Therefore, we can obtain multiple component learners with the diversity of the selected samples from different subsets. Algorithm 1 summarizes the pseudocode for producing component learners with bagging.

**Algorithm 1** Framework of Bagging

**Input:** $S$: Training set; $L$: Learner; $N^B$: Number of iterations
1: **for** $n = 1$ to $N^B$
2: $S_n =$ bootstrap sample from $S$.
3: $C_n^B = L(S_n)$
4: **end**
**Output:** Multiple component learners $\{C_n^B | n = 1, \cdots, N^B\}$.

**Algorithm 2** Framework of Random Subspace

**Input:** $F$: Feature set; $L$: Learner; $N^R$: Number of iterations
1: **for** $n = 1$ to $N^R$
2: $F_n =$bootstrap feature from $F$.
3: $C_n^R = L(F_n)$
4: **end**
**Output:** Multiple component learners $\{C_n^R | n = 1, \cdots, N^R\}$.

**Algorithm 3** Framework of Inclusive Subspace

**Input:** $F$: Feature set; $L$: Learner; $N^I$: Number of iterations
1: **for** $n = 1$ to $N^I$
2: $F_n = [F(1), \ldots, F(n+1)]$.
3: $C_n^I = L(F_n)$
4: **end**
**Output:** Multiple component learners $\{C_n^I | n = 1, \cdots, N^I\}$.

The random subspace method, an elaborate integration of bootstrapping and aggregation that is akin to bagging and enjoys the merits of both approaches [32], was applied to environmental factors. Compared to bagging, which bootstraps training samples, the difference of random subspace is that it exerts bootstrapping on the input features. Generally, with a high-dimensional feature vector or a small number of training samples, it is very likely that the problem of overfitting will be introduced. As presented in Fig. 3(a), directly using all 12 features is not always superior to using a portion of the features, which might be caused by overfitting. To settle this problem, a novel subset composed of a portion of the features is generated to decrease the low conformity between the size of the training samples and the length of the feature vector. Using the new subset, we can obtain a component learner. Repeating the above-mentioned process by random sampling applied to the feature space, we are able to establish multiple component learners with the diversity of the environmen-

tal factors. Algorithm 2 shows the pseudocode for creating component learners with the random subspace.

For temporal factors, this paper proposes a new method to generate diversity. It is apparent that the historical data that are closer to the current moment have a higher correlation with the $PM_{2.5}$ concentration at the next moment and thus make a greater contribution to the $PM_{2.5}$ prediction. On this basis, we consider the present time to be $T_0$, and the $PM_{2.5}$ concentration after $T_0$ is to be predicted. Therefore, we build multiple subsets, which, respectively, contain the $PM_{2.5}$ values recorded at the time of $\{T_0, T_{-1}\}$, $\{T_0, T_{-1}, T_{-2}\}$, $\{T_0, T_{-1}, T_{-2}, T_{-3}\}, \cdots$. Using these new subsets, we can derive multiple component learners with a diversity of temporal factors. Following the name of random subspace, we call this new method inclusive subspace since each subset is included in its latter subsets. Algorithm 3 shows the pseudocode for generating component learners with the inclusive subspace approach.

Second, we employ a pruning technique to delete negative component learners in each of three types. Directly aggregating all the component learners is not a good choice due to the existence of negative component learners that deteriorates the effectiveness of ensemble [30]. Supposing that a task is to use an ensemble that approximates a function $H : \mathbb{R}^s \to \mathbb{R}^t$, $z \in \mathbb{R}^s$ is sampled in light of a distribution $P(z)$, the expected output of $z$ is $\bar{z}$, and the real output of the $x$th component learner is $H_x(z)$, we can derive the output of the ensemble on $z$ is

$$\widehat{H}(z) = \sum_{x=1}^{r} \omega_x H_x(z) \tag{1}$$

where $r$ means the total number of component learners, $\omega_x \in [0, 1]$, and $\sum_{x=1}^{r} \omega_x = 1$. The generalization error $Err_x(z)$ of the $x$th component learner on $z$ and the generalization error $\widehat{Err}(z)$ of the ensemble on $z$ are defined as

$$Err_x(z) = (H_x(z) - \bar{z})^2, \tag{2}$$
$$\widehat{Err}(z) = (\widehat{H}(z) - \bar{z})^2. \tag{3}$$

Then, the generalization error of the $x$th component learner and that of the ensemble on $P(z)$ can be expressed by

$$Err_x = \int Err_x(z) P(z) dz \tag{4}$$
$$\widehat{Err} = \int \widehat{Err}(z) P(z) dz. \tag{5}$$

The correlation between the $x$th and $y$th component learners can be expressed by

$$Corr_{xy} = \int \sqrt{Err_x}\sqrt{Err_y} P(z) dz \tag{6}$$

where $Corr_{xy} = Corr_{yx}$ and $Corr_{xx} = Err_x$. Incorporating (2) and (4), we can derive

$$\widehat{Err}(z) = \left(\sum_{x=1}^{r} \omega_x H_x(z) - \bar{z}\right)\left(\sum_{y=1}^{r} \omega_y H_y(z) - \bar{z}\right). \tag{7}$$

Furthermore, combining (6)–(8), we are able to attain

$$\widehat{Err} = \sum_{x=1}^{r}\sum_{y=1}^{r} \omega_x \omega_y Corr_{xy}. \tag{8}$$

For convenience of analysis, we hypothesize that the overall component learners are of the equivalent contributions to the ensemble, or in other words, of the equal weights. Therefore, (8) can be rewritten as

$$\widehat{Err} = \frac{1}{r^2}\sum_{x=1}^{r}\sum_{y=1}^{r} Corr_{xy}. \tag{9}$$

Next, we attempt to exclude one certain component leaner, for instance, the $q$th component learner, from the ensemble and concentrate on whether this operation is beneficial to the final ensemble output. After removing the $q$th component learner, according to (2)–(10), we can derive the generalization error of the new ensemble as

$$\widehat{Err'} = \frac{1}{(r-1)^2}\sum_{\substack{x=1 \\ x\neq q}}^{r}\sum_{\substack{y=1 \\ y\neq q}}^{r} Corr_{xy}. \tag{10}$$

From (9) and (10), we deduce that the ensemble that deletes the $q$th component learner is superior to the one that contains the $q$th component learner if $\widehat{Err'}$ is lower than $\widehat{Err}$, namely,

$$\widehat{Err} \leqslant \frac{1}{2r-1}\left(2\sum_{\substack{x=1 \\ x\neq q}}^{r} Corr_{xq} + Err_q\right). \tag{11}$$

Moreover, we substitute (9) into (11) and make simplification

$$(2r-1)\sum_{x=1}^{r}\sum_{y=1}^{r} Corr_{xy} \leqslant 2r^2 \sum_{\substack{x=1 \\ x\neq q}}^{r} Corr_{xq} + r^2 Err_q. \tag{12}$$

As thus, we can make sure that the $q$th component learner is a negative component learner and should be excluded if its associated generalization error is larger than a threshold

$$Err_q \geqslant Thr_q = \frac{2r-1}{r^2}\sum_{x=1}^{r}\sum_{y=1}^{r} Corr_{xy} - 2\sum_{\substack{x=1 \\ x\neq q}}^{r} Corr_{xq}. \tag{13}$$

Based on the above-mentioned pruning criterion, we can delete negative component learners from the entire three categories of component learners. In traditional methods, each component learner is examined regarding whether or not it negatively impacts the ensemble compared to all the other component learners. Instead, in this work, we consider the following facts.

1) Each type of component learners (e.g., with the diverse selected samples) is considerably different from the other types of component learners (e.g. with the diverse environmental factors or temporal factors), and thereby, comparisons among them are unfair and lack sufficient reasons.
2) The majority or even all of the component learners in one certain type, due to their inferior performance, might be excluded, which directly deteriorates the diversity and

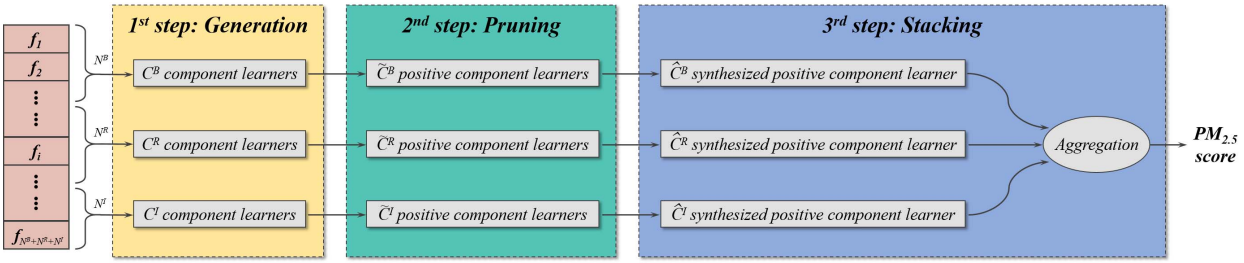



Fig. 4. Basic flowchart of the proposed SSEP predictive model.

indirectly decreases the effectiveness of ensemble. With these concerns, we use a discriminative approach of pruning that merely compares the component learners in the same category and removes those component learners that introduce a passive influence on the ensemble.

Third, we implement a stacking technique on the selected positive component learners to predict the future $PM_{2.5}$ concentration. Note that in the first step, we generate $N^B$ component learners with the diverse environmental factors, $N^R$ component learners with the diverse temporal factors, and $N^I$ component learners with the diverse selected samples. Next, in the second step, we use a pruning method to remove negative component learners in each type and maintain the positive component learners that comprise a positive component learner set $\widetilde{C} = [\widetilde{C}^B, \widetilde{C}^R, \widetilde{C}^I]^*$, where $\widetilde{C}^B \subset C^B$, $\widetilde{C}^R \subset C^R$, $\widetilde{C}^I \subset C^I$, and the superscript '*' means transpose. Afterward, a simple method may be leveraged to straightforwardly incorporate the selected positive component learners to infer the prediction scores of $PM_{2.5}$ concentration in the future.

Nonetheless, in practical Apps, it was found that the aforesaid straightforward incorporation-based method works poorly. According to the analyses on the problem of $PM_{2.5}$ prediction and the intermediate results, we claim that the poor performance is very likely caused by the subsequent two reasons. First, after the first and second steps, approximately 20%–50% of component learners in each type will be excluded, and thus, a considerable number of component learners will be still preserved. In this case, due to the "curse of dimensionality," directly merging such high-dimensional data can easily result in overfitting; therefore, it is not a beneficial approach to derive a $PM_{2.5}$ predictor with strong generalization ability. Second, most poorly performing component learners will be neglected if a straightforward fusion is used, which will inevitably be an opportunity lost for improving the ensemble's diversity. Overall, it is not effective and advisable to directly aggregate the selected positive component learners.

Based on the above-mentioned considerations, in this paper, we first conduct the direct average on those selected positive component learners in each category and derive three synthesized positive component learners, respectively, denoted as $\widehat{C}^B$, $\widehat{C}^R$, and $\widehat{C}^I$. For convenience, we define a new synthesized positive component learner set to be $\widehat{C} = [\widehat{C}^B, \widehat{C}^R, \widehat{C}^I]^*$. Next, we merge the three synthesized positive component learners to infer the $PM_{2.5}$ value

$$p = \mathbf{w}^* \phi(\mathbf{v}) + b \tag{14}$$

where $\mathbf{w}$ and $b$ are model parameters of weights and bias; $\phi(\cdot)$ is a function mapping the inputs into a high-dimensional feature space. We set $\mathbf{v}$ as $\widehat{C}$, i.e., $[v_1, v_2, v_3] = [\widehat{C}^B, \widehat{C}^R, \widehat{C}^I]$. Here, the following three strategies can be used to determine $\phi(\cdot)$, $\mathbf{w}$, and $b$.

1) *Direct Average:* We assign $\phi(\cdot)$ as the identity function, $b = 0$ and $\mathbf{w} = [1/3, 1/3, 1/3]^*$, and derive

$$p^{(a)} = \frac{1}{3}\sum_{h=1}^{3} v_h = \frac{1}{3}(\widehat{C}^B + \widehat{C}^R + \widehat{C}^I). \tag{15}$$

2) *Weighted Average:* We set $\phi(\cdot)$ as the identity function, $b = 0$ and $\mathbf{w} = [w_1, w_2, w_3]^* = (V_t^* V_t)^{-1} V_t^* \mathbf{p}_t$, where $V_t = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_u]^*$ and $\mathbf{p}_t = [p_1, p_2, \cdots, p_u]^*$ with $u$ is the total number of elements in the training set, $\mathbf{v}_i = [\widehat{C}_i^B, \widehat{C}_i^R, \widehat{C}_i^I]^* \in \mathbb{R}^3$ and $p_i \in \mathbb{R}^1$ are, respectively, the $i$th feature vector and the $i$th real target output in the training set, and derive

$$p^{(w)} = \mathbf{w}^* \phi(\mathbf{v}) = w_1 \widehat{C}^B + w_2 \widehat{C}^R + w_3 \widehat{C}^I. \tag{16}$$

3) *SVR-Based Regression:* We resort to solving the subsequent convex optimization function

$$\min_{\mathbf{w}, b, \boldsymbol{\zeta}, \boldsymbol{\zeta}'} \frac{1}{2}\|\mathbf{w}\|^2 + \kappa \sum_{i=1}^{u} (\zeta_i + \zeta_i')$$
$$\text{s. t.} \begin{cases} \mathbf{w}^* \phi(\mathbf{v}_i) + b - p_i \leq \epsilon + \zeta_i \\ p_i - \mathbf{w}^* \phi(\mathbf{v}_i) + b \leq \epsilon + \zeta_i' \\ \zeta_i, \zeta_i' \geqslant 0, \quad i = 1, 2, \cdots, u \end{cases} \tag{17}$$

where $\boldsymbol{\zeta}$ and $\boldsymbol{\zeta}'$ are a pair of slack variables as the margin of the errors; $\epsilon$ means the range of error tolerance; $\kappa$ is a positive regularization term for regulating the flatness of the function $p$ and tolerance limits of the error beyond $\epsilon$. The constraints above assure that most of the instances $\mathbf{v}_i$ are located in the tube $|\, p_i - \mathbf{w}^* \phi(\mathbf{v}_i) + b \,| \leqslant \epsilon$. Otherwise, an error $\boldsymbol{\zeta}$ or $\boldsymbol{\zeta}'$ will be generated and minimized in the objective function if $\mathbf{v}_i$ surpasses the tube. We can introduce Lagrangian multiplier $\mathbf{a}, \mathbf{a}', \boldsymbol{\mu}, \boldsymbol{\mu}' \geqslant 0$ and rewrite (17) by Lagrangian multiplier method

$$\begin{aligned} &L(\mathbf{w}, b, \mathbf{a}, \mathbf{a}', \boldsymbol{\zeta}, \boldsymbol{\zeta}', \boldsymbol{\mu}, \boldsymbol{\mu}') \\ &= \frac{1}{2}\|\mathbf{w}\|^2 + \kappa \sum_{i=1}^{u} (\zeta_i + \zeta_i') - \sum_{i=1}^{u} \mu_i \zeta_i - \sum_{i=1}^{u} \mu_i' \zeta_i' \\ &\quad + \sum_{i=1}^{u} a_i [c_i - \epsilon - \zeta_i] + \sum_{i=1}^{u} a_i' [-c_i - \epsilon - \zeta_i'] \end{aligned} \tag{18}$$

where $c_i = \mathbf{w}^*\phi(\mathbf{v}_i) + b - p_i$. We then let the partial derivative of $L(\mathbf{w}, b, \boldsymbol{\alpha}, \boldsymbol{\alpha}', \boldsymbol{\zeta}, \boldsymbol{\zeta}', \boldsymbol{\mu}, \boldsymbol{\mu}')$ with respect to $\boldsymbol{\alpha}, \boldsymbol{\alpha}', \boldsymbol{\zeta}, \boldsymbol{\zeta}'$ equal to zero and further define the kernel function $\mathcal{K}(\mathbf{v}_i, \mathbf{v}_j) = \phi(\mathbf{v}_i)^*\phi(\mathbf{v}_j)$ with the commonly used radial basis function (RBF) kernel, for mapping the data $\mathbf{v}$ to a higher dimensional space. After careful simplification and arrangement, we can derive

$$p^{(s)} = \sum_{i=1}^{u}(a_i' - a_i)\mathcal{K}(\mathbf{v}, \mathbf{v}_i) + \breve{b} \tag{19}$$

where $\breve{b} = p_i + \epsilon - \sum_{i=1}^{u}(a_i' - a_i)\mathbf{v}_i^*\mathbf{v}$.

Through experiments, we finally apply the SVR-based regression for stacking and assign $N^B = 10$, $N^R = 10$, and $N^I = 24$ in the proposed predictor. More comparisons and discussions concerning the selection of stacking technologies will be provided in Section III. In addition, we present a basic flowchart of our SSEP model in Fig. 4 for reader convenience: the first step is to generate multiple component learners based on bagging, random subspace, and inclusive subspace; the second step is to prune the negative component learners; and the third step is to apply stacking techniques to the selected positive component learners for aggregation.

## III. Experimental Results

This section is primarily devoted to examining the performance of the proposed SSEP model and comparing it with five popular predictors of $PM_{2.5}$ concentrations.

### A. Experimental Setup

The proposed SSEP is established to properly synthesize the component learners, which are generated by considering the diversities of environmental factors, temporal factors, and selected samples, to forecast the hourly $PM_{2.5}$ concentration over the following 48 h. In the first stage, our SSEP applies the methods of widely used sample selection, feature selection, etc., to produce component learners. Then, in the second stage, our SSEP applies the method of newly proposed component selection to derive the final output. For training and testing our SSEP model, we have gathered the hourly records of six air contaminations and six meteorological factors (including $PM_{2.5}$ concentration) from the 12 representative locations at Beijing, China, constituting 12 data sets, denoted as $D_1 \sim D_{12}$. We have successively collected the data during the last year. In this work, we apply the four typical used evaluation measures to check the effectiveness of the proposed SSEP predictor. In addition to the classical MSE, the other three evaluation indices are index of agreement (IA), normalized mean gross error (NMGE), and coefficient of determination ($R^2$).

1) The IA measures the difference between the predicted and observed values, as defined as follows:

$$\text{IA} = 1 - \frac{\sum_{i=1}^{n}(p_i - o_i)^2}{\sum_{i=1}^{n}(|p_i - \hat{o}| + |o_i - \hat{o}|)^2} \tag{20}$$

where $p_i$ and $o_i$ are the predicted and observed values of the $i$th sample, $\hat{o}$ is the mean of all $o_i$, and $n$ is the number of elements.

2) The NMGE indicates the mean error regardless of it is over or under estimation and is computed by

$$\text{NMGE} = \frac{\sum_{i=1}^{n}|p_i - o_i|}{\sum_{i=1}^{n} o_i}. \tag{21}$$

3) The $R^2$ reflects the linear relationship between predicted and observed values, as defined as follows:

$$R^2 = \frac{[\sum_{i=1}^{n}(p_i - \hat{p})(o_i - \hat{o})]^2}{\sum_{i=1}^{n}(p_i - \hat{p})^2 \sum_{i=1}^{n}(o_i - \hat{o})^2} \tag{22}$$

where $\hat{p}$ is the mean of all $p_i$.

Among these four criteria, a good predictive model should have a value close to 1 for IA and $R^2$, but a value close to 0 for MSE and NMGE.

According to the aforementioned four evaluation indices, we introduce five prevailing predictors of $PM_{2.5}$ concentration in the performance comparison. The first predictor, called the VOUK model [20], was proposed by applying principal component analysis followed by an artificial neural network. The second predictor, called the VLAC model [21], was designed based on a stepwise multiple linear regression. The third predictor, called the KABO model [22], was inspired by the adaptive neuro-fuzzy inference system. The fourth and fifth predictors, i.e., the Zheng and spatial-temporal deep neural network (ST-DNN) models [27], [28], respectively, were developed by properly incorporating multiple popular learners or neural networks. We remove the components related to spatial information in the Zheng and ST-DNN models since this work only concentrates on Beijing and does not involve different cities such as [27] and [28]; that is, each of 12 locations has highly similar latitude and longitude values. In subsequent comparisons, all models adopt the same condition when applied to predicting $PM_{2.5}$ concentration.

### B. Performance Evaluation

When comparing our proposed SSEP model with the five modern $PM_{2.5}$ predictive models, we first concentrate our attention on the $D_1$ data set and randomly classify it into three groups. One group contains 60% of the instances for training, the second group contains 20% of the instances for validation, and the third one contains the remaining 20% of the instances for testing. We repeat the above-mentioned process 200 times and compute the MSE, IA, NMGE, and $R^2$ values each time. The median values of MSE, IA, NMGE, and $R^2$ are derived and employed for comparison among the five $PM_{2.5}$ predictors tested. As listed in Table I, we illustrate the results of forecasting the $PM_{2.5}$ concentration on the $D_1$ data set at the moments of $T_1$, $T_2$, $T_3$, $T_4$, $T_5$, $T_6$, $T_9$, $T_{12}$, $T_{15}$, $T_{18}$, $T_{21}$, $T_{24}$, and $T_{48}$. As shown in Table I, the proposed SSEP model has achieved the best prediction performance in view of the four evaluation criteria above, superior to the other tested predictive models. In particular, we take the IA results of all the six predictors as an example. The relative performance gains between the proposed SSEP model and one of the five tested models are in the range of 0.1%–10.4% at $T_1$, 0.31%–10.6% at $T_2$, 0.32%–11.5% at $T_3$, 0.21%–13.0% at $T_4$, 0.33%–14.4% at $T_5$, 1.34%–18.5% at $T_6$, 0.69%–31.9%

TABLE I

COMPARISON OF LOG MSE, IA, NMGE, AND $R^2$ AMONG OUR SSEP AND FIVE MODERN MODELS. WE BOLD THE OPTIMAL ONE

| Index | Model | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ | $T_9$ | $T_{12}$ | $T_{15}$ | $T_{18}$ | $T_{21}$ | $T_{24}$ | $T_{48}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\log MSE$ | VOUK [20] | 3.373 | 3.436 | 3.497 | 3.540 | 3.575 | 3.624 | 3.718 | 3.769 | 3.776 | 3.792 | 3.819 | 3.827 | 3.848 |
| | VLAC [21] | 2.490 | 2.905 | 3.130 | 3.270 | 3.362 | 3.469 | 3.612 | 3.707 | 3.743 | 3.769 | 3.772 | 3.785 | 3.800 |
| | KABO [22] | 3.223 | 3.324 | 3.413 | 3.478 | 3.528 | 3.598 | 3.699 | 3.749 | 3.769 | 3.804 | 3.833 | 3.864 | 3.899 |
| | Zheng [27] | 2.517 | 2.919 | 3.149 | 3.225 | 3.318 | 3.408 | 3.513 | 3.557 | 3.563 | 3.584 | 3.603 | 3.617 | 3.647 |
| | ST-DNN [28] | 2.473 | 2.909 | 3.131 | 3.222 | 3.308 | 3.403 | 3.496 | 3.543 | 3.548 | 3.559 | 3.566 | 3.577 | 3.593 |
| | SSEP (Prop.) | **2.461** | **2.900** | **3.117** | **3.217** | **3.303** | **3.391** | **3.486** | **3.540** | **3.542** | **3.555** | **3.564** | **3.576** | **3.586** |
| $IA$ | VOUK [20] | 0.897 | 0.879 | 0.854 | 0.834 | 0.823 | 0.800 | 0.707 | 0.610 | 0.510 | 0.433 | 0.348 | 0.296 | 0.084 |
| | VLAC [21] | 0.989 | 0.969 | 0.949 | 0.927 | 0.910 | 0.880 | 0.810 | 0.716 | 0.623 | 0.570 | 0.506 | 0.478 | 0.362 |
| | KABO [22] | 0.913 | 0.887 | 0.858 | 0.830 | 0.807 | 0.763 | 0.661 | 0.568 | 0.464 | 0.355 | 0.341 | 0.333 | 0.329 |
| | Zheng [27] | 0.988 | 0.968 | 0.943 | 0.934 | 0.915 | 0.882 | 0.855 | 0.810 | 0.779 | 0.753 | 0.704 | 0.661 | 0.637 |
| | ST-DNN [28] | 0.989 | 0.969 | 0.949 | 0.936 | 0.920 | 0.892 | 0.866 | 0.822 | 0.791 | 0.774 | 0.748 | 0.705 | 0.683 |
| | SSEP (Prop.) | **0.990** | **0.972** | **0.952** | **0.938** | **0.923** | **0.904** | **0.872** | **0.831** | **0.804** | **0.798** | **0.792** | **0.760** | **0.731** |
| $NMGE$ | VOUK [20] | 0.348 | 0.363 | 0.388 | 0.406 | 0.416 | 0.443 | 0.520 | 0.574 | 0.627 | 0.693 | 0.737 | 0.763 | 0.799 |
| | VLAC [21] | 0.094 | 0.161 | 0.217 | 0.263 | 0.292 | 0.345 | 0.445 | 0.518 | 0.589 | 0.660 | 0.714 | 0.734 | 0.751 |
| | KABO [22] | 0.246 | 0.283 | 0.323 | 0.356 | 0.378 | 0.418 | 0.496 | 0.555 | 0.608 | 0.673 | 0.699 | 0.704 | 0.713 |
| | Zheng [27] | 0.102 | 0.166 | 0.224 | 0.250 | 0.279 | 0.316 | 0.373 | 0.429 | 0.480 | 0.503 | 0.546 | 0.586 | 0.625 |
| | ST-DNN [28] | 0.093 | 0.161 | 0.214 | 0.246 | 0.273 | 0.310 | 0.369 | 0.421 | 0.462 | 0.487 | 0.533 | 0.563 | 0.597 |
| | SSEP (Prop.) | **0.092** | **0.158** | **0.210** | **0.244** | **0.270** | **0.308** | **0.368** | **0.419** | **0.447** | **0.484** | **0.495** | **0.532** | **0.559** |
| $R^2$ | VOUK [20] | 0.682 | 0.636 | 0.580 | 0.547 | 0.523 | 0.473 | 0.351 | 0.243 | 0.155 | 0.102 | 0.071 | 0.050 | 0.001 |
| | VLAC [21] | 0.958 | 0.897 | 0.817 | 0.759 | 0.713 | 0.636 | 0.492 | 0.335 | 0.216 | 0.175 | 0.114 | 0.096 | 0.040 |
| | KABO [22] | 0.858 | 0.794 | 0.720 | 0.668 | 0.642 | 0.580 | 0.445 | 0.312 | 0.221 | 0.131 | 0.044 | 0.018 | 0.006 |
| | Zheng [27] | 0.958 | 0.895 | 0.818 | 0.791 | 0.737 | 0.674 | 0.614 | 0.579 | 0.462 | 0.444 | 0.412 | 0.362 | 0.282 |
| | ST-DNN [28] | 0.959 | 0.898 | 0.822 | 0.790 | 0.736 | 0.685 | 0.632 | 0.590 | 0.498 | 0.473 | 0.443 | 0.394 | 0.335 |
| | SSEP (Prop.) | 0.961 | 0.900 | 0.831 | 0.794 | 0.752 | 0.710 | 0.660 | 0.638 | 0.559 | 0.505 | 0.478 | 0.446 | 0.408 |

TABLE II

COMPARISON OF LOG MSE, IA, NMGE, AND $R^2$ AMONG THE SSEP PREDICTIVE MODELS BASED ON DIFFERENT TECHNOLOGIES

| Index | Model | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ | $T_9$ | $T_{12}$ | $T_{15}$ | $T_{18}$ | $T_{21}$ | $T_{24}$ | $T_{48}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\log MSE$ | SSEP_WE | 2.991 | 3.171 | 3.299 | 3.389 | 3.450 | 3.527 | 3.655 | 3.718 | 3.740 | 3.796 | 3.839 | 3.850 | 3.876 |
| | SSEP_SF | 2.865 | 3.104 | 3.260 | 3.352 | 3.422 | 3.505 | 3.634 | 3.706 | 3.731 | 3.786 | 3.835 | 3.846 | 3.871 |
| | SSEP_SI | 2.803 | 3.055 | 3.231 | 3.329 | 3.405 | 3.492 | 3.620 | 3.696 | 3.725 | 3.780 | 3.828 | 3.841 | 3.869 |
| | SSEP_DA | 2.776 | 2.996 | 3.159 | 3.242 | 3.320 | 3.406 | 3.503 | 3.586 | 3.610 | 3.625 | 3.640 | 3.665 | 3.720 |
| | SSEP_WA | 2.502 | 2.930 | 3.125 | 3.239 | 3.319 | 3.400 | 3.491 | 3.560 | 3.588 | 3.616 | 3.632 | 3.659 | 3.707 |
| | SSEP (Prop.) | 2.461 | 2.900 | 3.117 | 3.217 | 3.303 | 3.391 | 3.486 | 3.540 | 3.542 | 3.555 | 3.564 | 3.576 | 3.586 |
| $IA$ | SSEP_WE | 0.957 | 0.933 | 0.907 | 0.882 | 0.864 | 0.831 | 0.735 | 0.646 | 0.563 | 0.411 | 0.347 | 0.346 | 0.342 |
| | SSEP_SF | 0.970 | 0.946 | 0.919 | 0.897 | 0.880 | 0.850 | 0.767 | 0.680 | 0.596 | 0.453 | 0.363 | 0.354 | 0.345 |
| | SSEP_SI | 0.973 | 0.954 | 0.926 | 0.906 | 0.888 | 0.859 | 0.783 | 0.697 | 0.616 | 0.480 | 0.374 | 0.356 | 0.348 |
| | SSEP_DA | 0.976 | 0.958 | 0.935 | 0.921 | 0.907 | 0.884 | 0.843 | 0.773 | 0.720 | 0.691 | 0.648 | 0.581 | 0.527 |
| | SSEP_WA | 0.989 | 0.967 | 0.947 | 0.928 | 0.911 | 0.888 | 0.844 | 0.783 | 0.728 | 0.708 | 0.694 | 0.675 | 0.656 |
| | SSEP (Prop.) | 0.990 | 0.971 | 0.952 | 0.938 | 0.923 | 0.904 | 0.872 | 0.831 | 0.804 | 0.798 | 0.792 | 0.760 | 0.731 |
| $NMGE$ | SSEP_WE | 0.191 | 0.234 | 0.279 | 0.314 | 0.337 | 0.378 | 0.463 | 0.530 | 0.583 | 0.657 | 0.667 | 0.690 | 0.700 |
| | SSEP_SF | 0.166 | 0.216 | 0.264 | 0.298 | 0.322 | 0.363 | 0.449 | 0.519 | 0.572 | 0.649 | 0.663 | 0.684 | 0.697 |
| | SSEP_SI | 0.164 | 0.204 | 0.253 | 0.289 | 0.314 | 0.354 | 0.441 | 0.512 | 0.565 | 0.642 | 0.661 | 0.680 | 0.694 |
| | SSEP_DA | 0.150 | 0.200 | 0.243 | 0.269 | 0.286 | 0.329 | 0.394 | 0.461 | 0.514 | 0.569 | 0.591 | 0.611 | 0.635 |
| | SSEP_WA | 0.099 | 0.169 | 0.221 | 0.257 | 0.281 | 0.328 | 0.390 | 0.443 | 0.497 | 0.556 | 0.586 | 0.596 | 0.614 |
| | SSEP (Prop.) | 0.092 | 0.158 | 0.210 | 0.244 | 0.270 | 0.308 | 0.368 | 0.419 | 0.447 | 0.484 | 0.495 | 0.509 | 0.532 |
| $R^2$ | SSEP_WE | 0.899 | 0.829 | 0.751 | 0.696 | 0.664 | 0.604 | 0.465 | 0.327 | 0.236 | 0.150 | 0.053 | 0.025 | 0.010 |
| | SSEP_SF | 0.918 | 0.847 | 0.766 | 0.712 | 0.675 | 0.614 | 0.480 | 0.340 | 0.248 | 0.160 | 0.064 | 0.031 | 0.017 |
| | SSEP_SI | 0.928 | 0.859 | 0.776 | 0.722 | 0.685 | 0.621 | 0.488 | 0.350 | 0.257 | 0.170 | 0.077 | 0.035 | 0.023 |
| | SSEP_DA | 0.940 | 0.891 | 0.810 | 0.780 | 0.745 | 0.672 | 0.616 | 0.558 | 0.446 | 0.402 | 0.343 | 0.304 | 0.199 |
| | SSEP_WA | 0.958 | 0.892 | 0.813 | 0.784 | 0.750 | 0.683 | 0.625 | 0.572 | 0.467 | 0.428 | 0.384 | 0.352 | 0.260 |
| | SSEP (Prop.) | 0.961 | 0.900 | 0.831 | 0.794 | 0.752 | 0.710 | 0.660 | 0.638 | 0.559 | 0.505 | 0.478 | 0.446 | 0.408 |

at $T_9$, 1.09%–46.3% at $T_{12}$, 1.64%–73.3% at $T_{15}$, 3.10%–124% at $T_{18}$, 5.88%–132% at $T_{21}$, 7.80%–156 % at $T_{24}$, and 7.03%–770% at $T_{48}$. It is not difficult to determine that, on the one hand, our SSEP model is a better implementation than the other five popular predictive models, and on the other hand, the superiority of the proposed SSEP model dramatically increases from the short-term prediction (e.g., improved by 0.1%–10.4% at $T_1$) to the long-term prediction (e.g., improved by 7.03%–770% at $T_{48}$).

Note that in this paper, we introduce three methods for stacking: direct average, weighted average, and SVR-based regression. It is natural to perform a comparison between them that involves the proposed SSEP, SSEP_DA, and SSEP_WA (SSEP based on direct average and weighted average for

stacking followed by separately applying the selective ensemble to each of the three types of synthesized positive component learners that are generated by bagging, random subspace, and inclusive subspace). In addition to these three methods, we also include the SSEP_WE (SSEP without ensemble), SSEP_SF (SSEP with straightforward fusion after ensemble), and SSEP_SI (SSEP with straightforward incorporation after selective ensemble, which does not take the influence of different types of features into account) in the performance comparison. The results of the aforementioned six predictive models are reported in Table II. We can draw the subsequent two main conclusions as follows.

1) The proposed SSEP predictor has constantly delivered greater performance compared with other competing predictive models. More specifically, in terms of the index of IA, the relative performance gains between the SSEP model and the second-place predictive model in the five tested models are 0.10% at $T_1$, 0.41% at $T_2$, 0.53% at $T_3$, 1.08% at $T_4$, 1.31% at $T_5$, 1.80% at $T_6$, 3.32% at $T_9$, 6.13% at $T_{12}$, 10.4% at $T_{15}$, 12.7% at $T_{18}$, 14.1% at $T_{21}$, 12.6% at $T_{24}$, and 11.4% at $T_{48}$. Importantly, the superiority of our SSEP model becomes more apparent when observing its achievements that span the short-term prediction to the long-term prediction, which indicates the necessity of using bagging, random subspace, inclusive subspace, selective ensemble and stacking technologies.
2) According to the evaluation measures, we can derive the following rank: SSEP > SSEP_WA > SSEP_DA > SSEP_SI > SSEP_SF > SSEP_WE. As a natural extension, we further implement a comparison of the five pairs of models whose performance results are close to each other. First, we consider the difference of IA results between SSEP_WE and SSEP_SF at the $T_{48}$ moment. The relative performance gain of IA between these two models reaches up to 0.88%. This reflects that introducing ensemble learning is beneficial for boosting the prediction performance. Second, we calculate the relative performance improvement at $T_{48}$ between the SSEP_SI and SSEP_SF models. The gain exceeds 0.87%, and this result shows the positive contribution achieved by introducing the selective ensemble. Third, the SSEP_DA model is compared with SSEP_SI in terms of their IA values at $T_{48}$. The performance gain between them is unexpectedly higher than 51%, which illustrates that inserting the impacts of different types of features into the selective ensemble can greatly promote the performance of the $PM_{2.5}$ predictor. Fourth, we take the difference of IA results between SSEP_WA and SSEP_DA at the $T_{48}$ moment into account. The relative performance gain is approximately 24%, and this shows that there is a large gain by replacing the direct average with the weighted average. Fifth, we carry out a comparison on our proposed SSEP and SSEP_WA at $T_{48}$ and find a relative performance boost beyond 11.4%. It is obvious that the SVR-based regression leads to a considerable positive function on the forecast of $PM_{2.5}$ concentration in comparison with SSEP_WA, as well as SSEP_DA and other tested models. Overall, considering the differences among environmental factors, temporal factors, and selected samples, and the introduction of random subspace, inclusive subspace, bagging, selective ensemble, and stacking techniques, these methods play important roles in promoting the performance of the $PM_{2.5}$ predictor. Furthermore, appropriately combining the above-mentioned methods can result in a much greater performance gain compared with using parts of them.

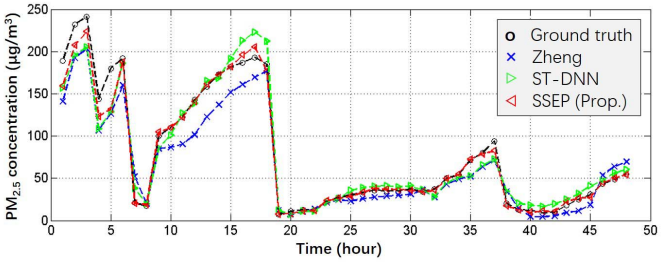



Fig. 5. Forecast of $PM_{2.5}$ concentration for a duration of 48 h.

### C. Comparison of Persistence Forecast

The comparison of the persistence forecast between our SSEP with the state-of-the-art Zheng and ST-DNN models is carried out. In this situation, we divide the entire $D_1$ data set into three sequential groups. More concretely, the training group contains the first 60% of the samples, the validation group contains the subsequent 20% of the samples, and the testing group contains the last 20% of the samples. We determine the aforementioned three models using the training and validation groups and then implement them to forecast the concentration of $PM_{2.5}$ for a duration of 48 h from the testing group. We show the ground-truth values and the associated prediction values of Zheng, ST-DNN, and SSEP in Fig. 5. Comparing these three testing models, it can be easily observed that the proposed SSEP outperforms the other two recently proposed predictive models. Furthermore, we can also find that the ST-DNN is superior to the Zheng model in many cases.

### D. Validation on Different Data Sets

More validations of the effectiveness of our SSEP predictor are also applied on the other 11 data sets. Similarly, the entire set of data samples is randomly divided into three groups with the ratio of 3:1:1, one each for training, validation, and testing. This process is repeatedly implemented 200 times to generate 200 values of MSE. We calculate their median values as illustrated in Tables III and IV. To facilitate a comparison, we highlight the best model across the ten tested predictive models with boldface. First, similar to the results on the $D_1$ data set, the proposed SSEP predictor always performs better than the other ten tested models on the $D_2$ to $D_{12}$ data sets, particularly for the long-term prediction. Second, it was found that the prediction performance of $PM_{2.5}$ concentration continually grows as the bagging, random subspace, inclusive

TABLE III

COMPARISON OF LOG MSE OF THE SSEP PREDICTOR AND TEN TESTED MODELS ON $D_2$–$D_7$ DATA SETS. WE BOLD THE OPTIMAL ONE

| Set | Model | $T_6$ | $T_{12}$ | $T_{18}$ | $T_{24}$ | $T_{48}$ |
|---|---|---|---|---|---|---|
| $D_2$ | VOUK [20] | 3.406 | 3.552 | 3.642 | 3.643 | 3.689 |
| | VLAC [21] | 3.287 | 3.535 | 3.635 | 3.637 | 3.675 |
| | KABO [22] | 3.392 | 3.570 | 3.713 | 3.725 | 3.729 |
| | Zheng [27] | 3.219 | 3.436 | 3.506 | 3.534 | 3.548 |
| | ST-DNN [28] | 3.216 | 3.426 | 3.497 | 3.527 | 3.540 |
| | SSEP_WE | 3.312 | 3.527 | 3.678 | 3.718 | 3.719 |
| | SSEP_SF | 3.290 | 3.516 | 3.666 | 3.712 | 3.716 |
| | SSEP_SI | 3.278 | 3.513 | 3.656 | 3.707 | 3.714 |
| | SSEP_DA | 3.235 | 3.440 | 3.526 | 3.556 | 3.574 |
| | SSEP_WA | 3.204 | 3.424 | 3.523 | 3.545 | 3.558 |
| | SSEP (Prop) | **3.202** | **3.402** | **3.486** | **3.501** | **3.514** |
| $D_3$ | VOUK [20] | 3.260 | 3.392 | 3.413 | 3.440 | 3.478 |
| | VLAC [21] | 3.162 | 3.384 | 3.420 | 3.443 | 3.481 |
| | KABO [22] | 3.279 | 3.396 | 3.459 | 3.490 | 3.520 |
| | Zheng [27] | 3.027 | 3.214 | 3.236 | 3.315 | 3.396 |
| | ST-DNN [28] | 3.022 | 3.212 | 3.233 | 3.309 | 3.386 |
| | SSEP_WE | 3.202 | 3.359 | 3.457 | 3.486 | 3.519 |
| | SSEP_SF | 3.177 | 3.348 | 3.452 | 3.484 | 3.518 |
| | SSEP_SI | 3.160 | 3.340 | 3.448 | 3.480 | 3.517 |
| | SSEP_DA | 3.058 | 3.238 | 3.270 | 3.328 | 3.410 |
| | SSEP_WA | 3.038 | 3.222 | 3.239 | 3.324 | 3.391 |
| | SSEP (Prop.) | **3.013** | **3.199** | **3.226** | **3.283** | **3.378** |
| $D_4$ | VOUK [20] | 3.589 | 3.708 | 3.757 | 3.774 | 3.793 |
| | VLAC [21] | 3.430 | 3.643 | 3.726 | 3.739 | 3.755 |
| | KABO [22] | 3.537 | 3.691 | 3.783 | 3.817 | 3.820 |
| | Zheng [27] | 3.314 | 3.494 | 3.544 | 3.588 | 3.593 |
| | ST-DNN [28] | 3.314 | 3.490 | 3.536 | 3.576 | 3.590 |
| | SSEP_WE | 3.467 | 3.653 | 3.756 | 3.811 | 3.817 |
| | SSEP_SF | 3.449 | 3.641 | 3.742 | 3.804 | 3.813 |
| | SSEP_SI | 3.437 | 3.634 | 3.733 | 3.800 | 3.812 |
| | SSEP_DA | 3.345 | 3.528 | 3.581 | 3.621 | 3.657 |
| | SSEP_WA | 3.318 | 3.512 | 3.545 | 3.580 | 3.613 |
| | SSEP (Prop.) | **3.312** | **3.486** | **3.531** | **3.570** | **3.573** |
| $D_5$ | VOUK [20] | 3.551 | 3.655 | 3.688 | 3.690 | 3.732 |
| | VLAC [21] | 3.412 | 3.643 | 3.692 | 3.703 | 3.731 |
| | KABO [22] | 3.519 | 3.646 | 3.705 | 3.737 | 3.765 |
| | Zheng [27] | 3.276 | 3.478 | 3.512 | 3.567 | 3.621 |
| | ST-DNN [28] | 3.276 | 3.472 | 3.507 | 3.564 | 3.604 |
| | SSEP_WE | 3.458 | 3.612 | 3.681 | 3.723 | 3.761 |
| | SSEP_SF | 3.429 | 3.599 | 3.673 | 3.717 | 3.759 |
| | SSEP_SI | 3.416 | 3.593 | 3.671 | 3.716 | 3.758 |
| | SSEP_DA | 3.291 | 3.497 | 3.533 | 3.581 | 3.639 |
| | SSEP_WA | 3.272 | 3.476 | 3.524 | 3.573 | 3.630 |
| | SSEP (Prop.) | **3.270** | **3.460** | **3.470** | **3.542** | **3.591** |
| $D_6$ | VOUK [20] | 3.413 | 3.568 | 3.677 | 3.689 | 3.694 |
| | VLAC [21] | 3.275 | 3.528 | 3.662 | 3.682 | 3.684 |
| | KABO [22] | 3.336 | 3.566 | 3.699 | 3.742 | 3.754 |
| | Zheng [27] | 3.233 | 3.438 | 3.491 | 3.497 | 3.510 |
| | ST-DNN [28] | 3.224 | 3.421 | 3.480 | 3.498 | 3.511 |
| | SSEP_WE | 3.287 | 3.526 | 3.671 | 3.730 | 3.809 |
| | SSEP_SF | 3.271 | 3.507 | 3.660 | 3.719 | 3.806 |
| | SSEP_SI | 3.264 | 3.498 | 3.655 | 3.713 | 3.804 |
| | SSEP_DA | 3.243 | 3.436 | 3.530 | 3.560 | 3.750 |
| | SSEP_WA | 3.230 | 3.426 | 3.523 | 3.541 | 3.561 |
| | SSEP (Prop.) | **3.223** | **3.411** | **3.475** | **3.488** | **3.505** |
| $D_7$ | VOUK [20] | 3.610 | 3.677 | 3.727 | 3.732 | 3.847 |
| | VLAC [21] | 3.447 | 3.617 | 3.697 | 3.713 | 3.828 |
| | KABO [22] | 3.582 | 3.645 | 3.731 | 3.764 | 3.884 |
| | Zheng [27] | 3.362 | 3.489 | 3.563 | 3.625 | 3.676 |
| | ST-DNN [28] | 3.358 | 3.483 | 3.540 | 3.612 | 3.649 |
| | SSEP_WE | 3.531 | 3.616 | 3.710 | 3.743 | 3.879 |
| | SSEP_SF | 3.506 | 3.604 | 3.704 | 3.737 | 3.875 |
| | SSEP_SI | 3.493 | 3.598 | 3.699 | 3.735 | 3.871 |
| | SSEP_DA | 3.372 | 3.513 | 3.580 | 3.623 | 3.730 |
| | SSEP_WA | 3.358 | 3.491 | 3.579 | 3.614 | 3.665 |
| | SSEP (Prop.) | **3.333** | **3.471** | **3.515** | **3.580** | **3.626** |

TABLE IV

COMPARISON OF LOG MSE OF THE SSEP PREDICTOR AND TEN TESTED MODELS ON $D_8$–$D_{12}$ DATA SETS. WE BOLD THE OPTIMAL ONE

| Set | Model | $T_6$ | $T_{12}$ | $T_{18}$ | $T_{24}$ | $T_{48}$ |
|---|---|---|---|---|---|---|
| $D_8$ | VOUK [20] | 3.320 | 3.485 | 3.539 | 3.555 | 3.602 |
| | VLAC [21] | 3.221 | 3.476 | 3.532 | 3.541 | 3.595 |
| | KABO [22] | 3.345 | 3.488 | 3.553 | 3.602 | 3.642 |
| | Zheng [27] | 3.049 | 3.281 | 3.363 | 3.449 | 3.502 |
| | ST-DNN [28] | 3.044 | 3.277 | 3.355 | 3.445 | 3.506 |
| | SSEP_WE | 3.252 | 3.456 | 3.526 | 3.587 | 3.638 |
| | SSEP_SF | 3.220 | 3.442 | 3.516 | 3.580 | 3.638 |
| | SSEP_SI | 3.202 | 3.437 | 3.505 | 3.574 | 3.638 |
| | SSEP_DA | 3.062 | 3.271 | 3.378 | 3.421 | 3.496 |
| | SSEP_WA | 3.047 | 3.253 | 3.377 | 3.399 | 3.492 |
| | SSEP (Prop) | **3.039** | **3.234** | **3.350** | **3.380** | **3.447** |
| $D_9$ | VOUK [20] | 3.517 | 3.639 | 3.717 | 3.740 | 3.767 |
| | VLAC [21] | 3.422 | 3.630 | 3.712 | 3.721 | 3.749 |
| | KABO [22] | 3.528 | 3.660 | 3.738 | 3.809 | 3.854 |
| | Zheng [27] | 3.320 | 3.467 | 3.541 | 3.595 | 3.678 |
| | ST-DNN [28] | 3.314 | 3.466 | 3.542 | 3.583 | 3.676 |
| | SSEP_WE | 3.455 | 3.623 | 3.700 | 3.790 | 3.831 |
| | SSEP_SF | 3.431 | 3.611 | 3.689 | 3.776 | 3.810 |
| | SSEP_SI | 3.417 | 3.603 | 3.681 | 3.766 | 3.800 |
| | SSEP_DA | 3.333 | 3.479 | 3.565 | 3.602 | 3.684 |
| | SSEP_WA | 3.329 | 3.459 | 3.545 | 3.596 | 3.680 |
| | SSEP (Prop.) | **3.308** | **3.453** | **3.524** | **3.543** | **3.639** |
| $D_{10}$ | VOUK [20] | 3.493 | 3.591 | 3.596 | 3.609 | 3.618 |
| | VLAC [21] | 3.360 | 3.576 | 3.578 | 3.586 | 3.587 |
| | KABO [22] | 3.548 | 3.615 | 3.630 | 3.651 | 3.674 |
| | Zheng [27] | 3.275 | 3.420 | 3.451 | 3.485 | 3.488 |
| | ST-DNN [28] | 3.286 | 3.425 | 3.452 | 3.481 | 3.483 |
| | SSEP_WE | 3.459 | 3.604 | 3.610 | 3.614 | 3.618 |
| | SSEP_SF | 3.417 | 3.598 | 3.600 | 3.602 | 3.616 |
| | SSEP_SI | 3.393 | 3.588 | 3.597 | 3.598 | 3.606 |
| | SSEP_DA | 3.299 | 3.459 | 3.483 | 3.500 | 3.512 |
| | SSEP_WA | 3.279 | 3.449 | 3.462 | 3.479 | 3.490 |
| | SSEP (Prop.) | **3.264** | **3.411** | **3.442** | **3.476** | **3.477** |
| $D_{11}$ | VOUK [20] | 3.367 | 3.549 | 3.647 | 3.681 | 3.870 |
| | VLAC [21] | 3.251 | 3.536 | 3.623 | 3.678 | 3.757 |
| | KABO [22] | 3.341 | 3.555 | 3.685 | 3.757 | 3.804 |
| | Zheng [27] | 3.196 | 3.398 | 3.486 | 3.519 | 3.584 |
| | ST-DNN [28] | 3.261 | 3.452 | 3.497 | 3.524 | 3.574 |
| | SSEP_WE | 3.284 | 3.524 | 3.651 | 3.735 | 3.795 |
| | SSEP_SF | 3.268 | 3.512 | 3.635 | 3.725 | 3.790 |
| | SSEP_SI | 3.261 | 3.506 | 3.625 | 3.718 | 3.789 |
| | SSEP_DA | 3.189 | 3.398 | 3.518 | 3.559 | 3.634 |
| | SSEP_WA | 3.172 | 3.382 | 3.509 | 3.510 | 3.596 |
| | SSEP (Prop.) | **3.166** | **3.372** | **3.467** | **3.480** | **3.558** |
| $D_{12}$ | VOUK [20] | 3.549 | 3.702 | 3.726 | 3.735 | 3.884 |
| | VLAC [21] | 3.432 | 3.680 | 3.707 | 3.721 | 3.868 |
| | KABO [22] | 3.529 | 3.714 | 3.766 | 3.809 | 3.911 |
| | Zheng [27] | 3.356 | 3.560 | 3.569 | 3.648 | 3.703 |
| | ST-DNN [28] | 3.352 | 3.558 | 3.565 | 3.633 | 3.696 |
| | SSEP_WE | 3.469 | 3.680 | 3.737 | 3.802 | 3.911 |
| | SSEP_SF | 3.447 | 3.668 | 3.725 | 3.798 | 3.911 |
| | SSEP_SI | 3.438 | 3.663 | 3.718 | 3.792 | 3.911 |
| | SSEP_DA | 3.374 | 3.588 | 3.608 | 3.653 | 3.769 |
| | SSEP_WA | 3.369 | 3.575 | 3.600 | 3.651 | 3.731 |
| | SSEP (Prop.) | **3.350** | **3.547** | **3.558** | **3.611** | **3.680** |

subspace, selective ensemble, and stacking technologies are introduced.

### *E. Discussion*

Although our SSEP has attained the highest performance among all the competing models used in this paper, the gain between our SSEP and the state-of-the-art ST-DNN is not always large. We further compare the proposed SSEP with the

ST-DNN and derive some conclusions as follows. First, due to the use of various categories of high-complexity deep neural networks, the ST-DNN needs more computational resources. In comparison, because the proposed SSEP only depends on the SVR, it requires relatively less implementation costs. Second, because of the SVR's resistance to overfitting, our proposed SVR-based SSEP should be more robust than the ST-DNN based on deep neural networks. Third, considering the merit of deep neural networks in mining deeper relationships than popular machine learners, such as SVR, a good research direction would be an integration of the proposed stacked selective ensemble method with deep neural networks.

## IV. Conclusion

In this paper, we have explored the problem of forecasting $PM_{2.5}$ concentrations. According to observations and analyses from recorded data, the three types of features, namely, environmental factors, temporal factors, and selected samples, all play important roles. Furthermore, applying a proper combination of these three types of features can lead to remarkable benefits in predicting the $PM_{2.5}$ concentration. With the above-mentioned considerations, this paper has designed the stacked selective ensemble-based predictor (SSEP) by introducing the technologies of random subspace, inclusive subspace, bagging, selective ensemble, SVR-based stacking, and so on. Extensive experiments conducted on the 12 data sets demonstrate that our proposed SSEP model is prominently superior to the state-of-the-art competitors and intermediate models that exploit part of the aforementioned technologies used in the SSEP predictor.

In this paper, we focus on the prediction of $PM_{2.5}$ concentration in the city of Beijing. Future studies will be devoted to other cities by introducing the following two strategies. One strategy is to transfer the proposed SSEP model to a model with modified ensemble methods, such as selectively incorporating bagging, random subspace, and inclusive subspace according to the specific characteristic of a city. The other strategy is to exploit multitask learning to share knowledge from the proposed predictor with other cities.

## References


[1] Y. Song *et al.*, "Source apportionment of $PM_{2.5}$ in Beijing by positive matrix factorization," *Atmos. Environ.*, vol. 40, no. 8, pp. 1526–1537, Mar. 2006.

[2] B. Lv, Y. Hu, H. H. Chang, A. G. Russell, and Y. Bai, "Improving the accuracy of daily $PM_{2.5}$ distributions derived from the fusion of ground-level measurements with aerosol optical depth observations, a case study in north China," *Environ. Sci. Technol.*, vol. 50, no. 9, pp. 4752–4759, Apr. 2016.

[3] K. Gu, J. Qiao, and X. Li, "Highly efficient picture-based prediction of $PM_{2.5}$ concentration," *IEEE Trans. Ind. Electron.*, vol. 66, no. 4, pp. 3176–3184, Apr. 2019.

[4] G. Yue, K. Gu, and J. Qiao, "Effective and efficient photo–based $PM_{2.5}$ concentration estimation," *IEEE Trans. Instrum. Meas.*, to be published. doi: 10.1109/TIM.2018.2886091.

[5] *Standard Operating Procedure for Particulate Matter (PM) Gravimetric Analysis*. Accessed: Jan. 15, 2019. [Online]. Available: https://www3.epa.gov/ttnamti1/files/ambient/pm25/spec/RTIGravMassSOPFINAL.pdf

[6] *$PM_{2.5}$ and $PM_{10}$ Beta Attenuation Monitor Operating Procedure*. Accessed: Jan. 15, 2019. [Online]. Available: https://fortress.wa.gov/ecy/publications/documents/1702005.pdf

[7] K. L. Demerjian, "The mechanism of photochemical smog formation," *Adv. Environ. Sci. Technol.*, vol. 4, pp. 1–262, 1974.

[8] J. H. Seinfeld and S. N. Pandis, *Atmospheric Chemistry and Physics: From Air Pollution to Climate Change*. Hoboken, NJ, USA: Wiley, 2016.

[9] M. W. Gerry, G. Z. Whitten, J. P. Killus, and M. C. Dodge, "A photochemical mechanism for urban and regional scale computer modelling," *J. Geophys. Res., Atmos.*, vol. 94, pp. 12925–12956, 1989.

[10] W. R. Stockwell, F. Kirchner, M. Kuhn, and S. Seefeld, "A new mechanism for regional atmospheric chemistry modeling," *J. Geophys. Res., Atmos.*, vol. 102, no. D22, pp. 25847–25879, 1997.

[11] R. Lu, R. P. Turco, and M. Z. Jacobson, "An integrated air pollution modeling system for urban and regional scales: 1. Structure and performance," *J. Geophys. Res., Atmos.*, vol. 102, no. D5, pp. 6063–6079, 1997.

[12] R. Lu, R. P. Turco, and M. Z. Jacobson, "An integrated air pollution modeling system for urban and regional scales: 2. Simulations for SCAQS 1987," *J. Geophys. Res., Atmos.*, vol. 102, no. D5, pp. 6081–6098, 1997.

[13] C.-H. Huang and C.-Y. Tai, "Relative humidity effect on $PM_{2.5}$ readings recorded by collocated beta attenuation monitors," *Environ. Eng. Sci.*, vol. 25, no. 7, pp. 1079–1090, Sep. 2008.

[14] J. Wang and S. Ogawa, "Effects of meteorological conditions on $PM_{2.5}$ concentrations in Nagasaki, Japan," *Int. J. Environ. Res. Public Health*, vol. 12, no. 8, pp. 9089–9101, Aug. 2015.

[15] J. B. Ordieres, E. P. Vergara, R. S. Capuz, and R. E. Salazar, "Neural network prediction model for fine particulate matter ($PM_{2.5}$) on the US–Mexico border in El Paso (Texas) and Ciudad Juárez (Chihuahua)," *Environ. Model. Softw.*, vol. 20, no. 5, pp. 547–559, May 2005.

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

[17] Y. Wang, Q. Ying, J. Hu, and H. Zhang, "Spatial and temporal variations of six criteria air pollutants in 31 provincial capital cities in China during 2013–2014," *Environ. Int.*, vol. 73, pp. 413–422, Dec. 2014.

[18] M. Elbayoumi, N. A. Ramli, N. F. F. Yusof, A. S. B. Yahaya, W. Al Madhoun, and A. Z. Ul-Saufie, "Multivariate methods for indoor $PM_{10}$ and $PM_{2.5}$ modelling in naturally ventilated schools buildings," *Atmos. Environ.*, vol. 94, pp. 11–21, Sep. 2014.

[19] V. Valverde, M. T. Pay, and J. M. Baldasano, "Circulation-type classification derived on a climatic basis to study air quality dynamics over the Iberian Peninsula," *Int. J. Climatol.*, vol. 35, no. 10, pp. 2877–2897, Aug. 2015.

[20] D. Voukantsis, K. Karatzas, J. Kukkonen, T. Räsänen, A. Karppinen, and M. Kolehmainen, "Intercomparison of air quality data using principal component analysis, and forecasting of $PM_{10}$ and $PM_{2.5}$ concentrations using artificial neural networks, in Thessaloniki and Helsinki," *Sci. Total Environ.*, vol. 409, no. 7, pp. 1266–1276, Mar. 2011.

[21] A. Vlachogianni, P. Kassomenos, A. Karppinen, S. Karakitsios, and J. Kukkonen, "Evaluation of a multiple regression model for the forecasting of the concentrations of $NO_x$ and $PM_{10}$ in Athens and Helsinki," *Sci. Total Environ.*, vol. 409, no. 8, pp. 1559–1571, Mar. 2011.

[22] S. Kaboodvandpour, J. Amanollahi, S. Qhavami, and B. Mohammadi, "Assessing the accuracy of multiple regressions, ANFIS, and ANN models in predicting dust storm occurrences in Sanandaj, Iran," *Natural Hazards*, vol. 78, no. 2, pp. 879–893, Apr. 2015.

[23] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[24] Accessed: Jul. 30, 2018. [Online]. Available: https://aqicn.org/city/beijing/us-embassy/cn/

[25] Accessed: Jul. 30, 2018. [Online]. Available: http://www.weather.com.cn/weather/101010100.shtml

[26] X. J. Zhao *et al.*, "Analysis of a winter regional haze event and its formation mechanism in the North China Plain," *Atmos. Chem. Phys.*, vol. 13, no. 11, pp. 5685–5696, Jun. 2013.

[27] Y. Zheng *et al.*, "Forecasting fine-grained air quality based on big data," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2015, pp. 2267–2276.

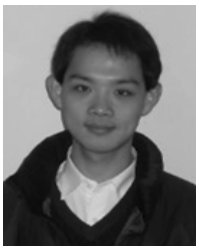
[28] P.-W. Soh, J.-W. Chang, and J.-W. Huang, "Adaptive deep learning-based air quality prediction model using the most relevant spatial-temporal relations," *IEEE Access*, vol. 6, pp. 38186–38199, 2018.

[29] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, Apr. 2011, Art. no. 27.

[30] Z.-H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: Many could be better than all," *Artif. Intell.*, vol. 137, nos. 1–2, pp. 239–263, May 2002.

[31] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.

[32] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, Aug. 1998.
[33] Z. Liu, H. Wang, R. Dollevoet, Y. Song, A. Núñez, and J. Zhang, "Ensemble EMD-based automatic extraction of the catenary structure wavelength from the pantograph–catenary contact force," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 10, pp. 2272–2283, Oct. 2016.
[34] E. Alickovic and A. Subasi, "Ensemble SVM method for automatic sleep stage classification," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 6, pp. 1258–1265, Jun. 2018.
[35] B. Uzunoğlu and M. A. Ülker, "Maximum likelihood ensemble filter state estimation for power systems," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 9, pp. 2097–2106, Sep. 2018.
[36] X. Wang and X. Tang, "Random sampling for subspace face recognition," *Int. J. Comput. Vis.*, vol. 70, no. 1, pp. 91–104, Oct. 2006.
[37] Z. Li, U. Park, and A. K. Jain, "A discriminative model for age invariant face recognition," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 1028–1037, Sep. 2011.
[38] G. Yue, C. Hou, K. Gu, N. Ling, and B. Li, "Analysis of structural characteristics for quality assessment of multiply distorted images," *IEEE Trans. Multimedia*, vol. 20, no. 10, pp. 2722–2732, Oct. 2018.
[39] J. Xia, M. D. Mura, J. Chanussot, P. Du, and X. He, "Random subspace ensembles for hyperspectral image classification with extended morphological attribute profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 9, pp. 4768–4786, Sep. 2015.
[40] R. Hang, Q. Liu, H. Song, and Y. Sun, "Matrix-based discriminant subspace ensemble for hyperspectral image spatial–spectral feature fusion, *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 2, pp. 783–794, Feb. 2016.



**Ke Gu** received the B.S. and Ph.D. degrees in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2009 and 2015, respectively.

He is currently a Professor with the Beijing University of Technology, Beijing, China. His current research interests include environmental perception, image processing, quality assessment, and machine learning.

Dr. Gu was the Leading Special Session Organizer in the VCIP 2016 and the ICIP 2017. He serves as a Guest Editor for the *Digital Signal Processing* (DSP). He is currently an Area Editor for *Signal Processing Image Communication* (SPIC). He was a recipient of the Best Paper Award from the IEEE TRANSACTIONS ON MULTIMEDIA (T-MM), the Best Student Paper Award at the IEEE International Conference on Multimedia and Expo (ICME) in 2016, and the Excellent Ph.D. Thesis Award from the Chinese Institute of Electronics (CIE) in 2016. He was an Associate Editor for the IEEE ACCESS and the *IET Image Processing* (IET-IP). He is a reviewer for 20 top SCI journals.

**Zhifang Xia** received the B.S. degree in measuring and control instrument from Anhui University, Hefei, China, in 2008, and the master's degree in control science and engineering from Tsinghua University, Beijing, China, in 2012.

She is currently an Engineer and a Registered Consultant (investment) with State Information Center, Beijing. Her current research interests include image processing, quality assessment, machine learning, and e-government.

Ms. Xia was a recipient of the second prize of National Excellent Engineering Consultation Award in 2016.

**Junfei Qiao** (M'11) received the B.E. and M.E. degrees in control engineer from Liaoning Technical University, Fuxin, China, in 1992 and 1995, respectively, and the Ph.D. degree from Northeast University, Shenyang, China, in 1998.

From 1998 to 2000, he was a Post-Doctoral Fellow with the School of Automatics, Tianjin University, Tianjin, China. He joined the Beijing University of Technology, Beijing, China, where he is currently a Professor. He is also the Director with the Intelligence Systems Laboratory, Beijing. His current research interests include neural networks, intelligent systems, self-adaptive/learning systems, and process control systems.

Dr. Qiao is a member of the IEEE Computational Intelligence Society. He is a reviewer for more than 20 international journals such as the IEEE TRANSACTIONS ON FUZZY SYSTEMS and the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS.