Using Ensemble Machine Learning Methods to Forecast Particulate Matter ($PM_{2.5}$) in Bangkok, Thailand

Patchanok Srisuradetchai(✉) and Wararit Panichkitkosolkul

Department of Mathematics and Statistics, Faculty of Science and Technology, Thammasat University, Pathum Thani 12120, Thailand
{patchanok,wararit}@mathstat.sci.tu.ac.th

Abstract. Many areas of Bangkok and its environs are currently blanketed with fine dust with dangerous levels of $PM_{2.5}$. High levels of $PM_{2.5}$ have a negative impact on human health. In this study, support vector regression, begged regression tree, random forest, gradient boosted models, neural networks, neural networks autoregressive, seasonal autoregressive moving average with exogenous covariates, k-nearest neighbor, Bayesian additive model, Prophet, and general additive models are used to anticipate $PM_{2.5}$. The usefulness of adopting an ensemble model for forecasting is investigated. A thorough evaluation of standalone algorithms and ensemble techniques was performed using the root-mean-square error, mean absolute error, and Pearson correlation coefficient. According to the results, hybrid models are effective in the forecasting of $PM_{2.5}$ concentrations.

Keywords: Ensemble machine learning · Predictive performance · $PM_{2.5}$

1 Introduction

All life on Earth depends on air, making it one of the most essential components. Over the last half-century, human activity, industry, automobiles, power plants, chemical plants, and other natural phenomena like agricultural burning, earthquakes, and fires have all contributed to an increase in pollution [1]. It is the fourth most common cause of death in the world. About 6.67 million people died in 2019 as a result of pollution, the most lethal of which scientists call $PM_{2.5}$ (particles measuring less than 2.5 μm in aerodynamic diameter) [2]. Breathing in $PM_{2.5}$ is harmful because it can reach the bloodstream and the lungs. For 24-h exposure to $PM_{2.5}$ both outdoors and indoors, most studies indicate that $PM_{2.5}$ of 12 $\mu g/m^3$ (micrograms per cubic meter) or less is considered healthy. Asthmatics and people who already have respiratory problems, such as those who live in areas with high levels of ozone in the air, should seek medical assistance if their symptoms worsen [3]. $PM_{2.5}$ has been related to an increase in respiratory and cardiovascular hospitalizations, emergency room visits, and mortality. Exposure to fine particles for a long time has been linked to chronic bronchitis, poor lung function, and death from lung cancer and heart disease [4].

O. Surinta and K. Kam Fung Yuen (Eds.): MIWAI 2022, LNAI 13651, pp. 204–215, 2022.
https://doi.org/10.1007/978-3-031-20992-5_18

Many areas of Bangkok and its environs are currently shrouded in fine dust with lethal $PM_{2.5}$ levels. According to the Pollution Control Department (PCD), the maximum allowable level of air dust in the environment is 50 $\mu g/m^3$, and this limit was exceeded numerous times in February 2022, with $PM_{2.5}$ levels ranging from 40 to 146 $\mu g/m^3$ [5]. Due to poor air quality, the PCD has expanded its list of Bangkok province areas and districts where people should wear protective gear and stay indoors.

Emissions from a variety of sources constantly change the quality of the air we breathe. As a result, things can appear normal one day and then take an unexpected turn the next. This emphasizes the importance of developing a tool for forecasting future air quality accurately. Machine learning techniques like artificial neural networks and regression trees can make predictions faster, more accurate, and easier to do with more than one type of data.

To forecast $PM_{2.5}$ in this paper, we utilize techniques of support vector regression (SVR) with linear and polynomial (degrees of 2 and 3) kernel functions, bagged regression trees (BRT), random forest (RF), gradient boosted models (GBM) with different loss functions, extreme gradient boosted (XGBoost) trees using L2 regularization, artificial neural networks (ANN) having 1 and 2 hidden layers and different activation functions, neural network autoregressive (NNAR), ANN using model averaging, seasonal ARIMAX (SARIMAX), k-nearest neighbor (KNN) regression with Epanechnikov and rectangular kernels, Prophet model, boosted generalized additive model (Boosted GAM), and Bayesian additive model (BAM). Finally, the "great" models will be included in the ensemble models to better forecast $PM_{2.5}$ concentration, and they will be compared to standalone algorithms. Following this introduction, the structure of the study is as follows: literature review; dataset overview and preparation; research methods; findings and conclusions, accordingly.

2 Literature Review

Because of the volatile nature of $PM_{2.5}$, accurate prediction has become challenging. Several models for predicting particulate matter emissions have been developed in the last several years in an effort to monitor air quality around the world.

Catalano [6] looked at the relationship between the hourly mean NO_2 concentration and factors that explain the NO_2 mean level one hour before, as well as traffic and weather conditions like the number of cars on the road, the speed of the wind, the direction of the wind, and the temperature. To model pollution peaks, the ANN, ARIMAX, and SARIMAX models were used.

Masood and Ahmad [7] looked at the possibilities of ANN and SVM in creating reliable and accurate $PM_{2.5}$ predictions for New Delhi. Carbon monoxide (CO), sulfur dioxide (SO_2), nitrogen oxide (NO), toluene (C_7H_8), nitrogen dioxide (NO_2), wind speed, relative humidity, and temperature are studied.

Suleiman et al. [8] evaluated and compared three air quality management techniques for predicting and managing roadside PM_{10} and $PM_{2.5}$, including SVM, ANN, and BRT. It has been found that the ANN and regression tree-based models perform marginally better than the SVM model for PM_{10} forecasting but significantly worse for $PM_{2.5}$ forecasting.

Doreswamy et al. [9] employed RF, XG Boost, BRT, and MLP Regression to forecast Taiwan $PM_{2.5}$. In both training and testing datasets, XG Boost obtained the best R^2 and lowest MSE. Sharma et al. [10] evaluated many air contaminants and utilized a time series regression model with extraneous factors to predict SO_2, NO_2, O_3, CO, and $PM_{2.5}$.

Qiao et al. [11] proposed a new model based on WT (wavelet transform)-SAE (stacked autoencoder)-LSTM (wavelet transform-stacked autoencoder-LSTM). To begin, WT is used to break down the $PM_{2.5}$ time series into numerous low- and high-frequency components based on different data from six Chinese research sites. SAE-LSTM is then used to forecast the deconstructed components.

Biancofiore et al. [12] analyzed three years of continuous measurements of PM and CO concentration in central Italy using a multiple linear regression model and ANN models with and without recursive architecture. One to three days in the future, an ANN was used to predict the concentrations of PM_{10} and $PM_{2.5}$ in the air. The ANN used meteorological and chemical factors as input.

Mahajan et al. [13] utilized the NNAR, an additive version of the Holt-Winters method, and the ARIMA model to forecast hourly $PM_{2.5}$ in Taiwan. For comparison, root-mean-square error (RMSE) and mean absolute error (MAE) were the criteria. The results show that the NNAR model has the lowest values for both RMSE and MAE.

Ejohwomu et al. [14] used ARIMA, exponential smoothing, prophet, NNAR, ANN based on multiple variables, SVM, XG Boost, and RF. $PM_{2.5}$, relative humidity, and temperature were measured every 15 min and converted to hourly time-series data. Accuracy of forecast model predictions was evaluated using metrics such as RMSE and MAE.

Gupta et al. [15] used NASA's Modern-Era Retrospective analysis for Research and Applications, Version 2 (MERRA2) aerosols and meteorology reanalysis data to estimate the surface $PM_{2.5}$ concentration in Thailand. The RF was used to validate and train the data. Furthermore, the RF can estimate hourly and daily mean PM2.5 with a high degree of precision. The mean bias is near to zero, with correlation coefficients above 0.90 in the majority of cases.

3 Dataset Overview and Preparation

The $PM_{2.5}$ air pollution index (API) data utilized in this study are secondary data obtained from the website of the World Air Quality Index (WAQI) project, https://aqicn.org/city/Bangkok/. Bangkok's API database in WAQI is retrieved from the Division of Air Quality Data, Bureau of Air Quality and Noise Management, Pollution Control Department. The daily AQI is based on the 24-h average of hourly readings from all stations from January 1, 2019 to December 31, 2021. Also, some variables are obtained from the World Meteorological Organization (WMO) via the webpage https://meteostat.net/en/place/th/bangkok. The database contains pollutants and meteorological variables such as ozone (O_3), nitrogen dioxide (NO_2), average temperature (Temp.avg), precipitation (PRCP), wind speed (Wspd), and pressure.

In data cleaning and preparation processes, the missing data can be accessed in aggregation plots as shown in Fig. 1. The missing proportion of PRCP is 1.09%, and $PM_{2.5}$, NO_2, and O_3 are all the same, at 0.36%, while the other covariates have no missing values. Two different types of blocks are shown on the right of Fig. 1: one for observed (blue) values and one for missing (red) data. There are 1,078 days with complete covariates, 12 days with only missing PRCP, 2 days with only missing $PM_{2.5}$, 2 days with missing NO_2 and O_3, and 2 days with missing $PM_{2.5}$, NO_2, and O_3. Multiple imputation by chained equations (MICE) was utilized to replace missing values in the gathered data. The MICE approach is based on the premise that multiple imputation is best accomplished in distinct steps, each of which may require diagnostic examination. Multiple imputation, analysis of imputed data, and pooling of analysis outcomes are MICE's main steps. This task can be achieved by using the R package "mice" [16].

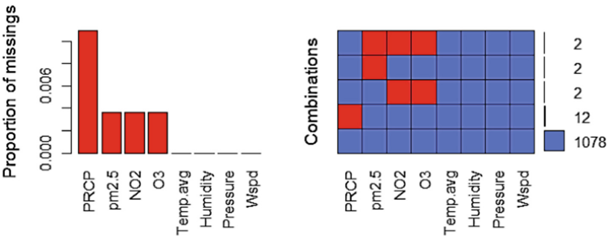

Fig. 1. Aggregation plots for missing values

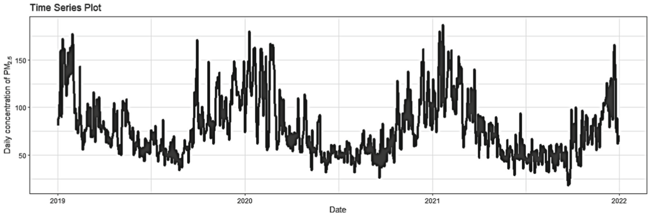

Fig. 2. Time-series plot of $PM_{2.5}$

Table 1. Descriptive statistics of cleaned data from 1 January 2019 to 31 December 2021

Covariates	Unit	Range	Mean	S.D	Min	Max
$PM_{2.5}$	$\mu g/m^3$	169.0	78.84	31.927	18.0	187.0
O3	$\mu g/m^3$	93.0	12.79	7.114	1.0	94.0
NO_2	$\mu g/m^3$	37.0	9.886	6.109	1.0	38.0
Temperature	Celsius	12.9	29.28	1.736	21.6	34.5
Precipitation	inches	117.1	3.967	10.625	0.0	117.1
Humidity	percent	51.8	71.19	9.063	44.0	95.8
Pressure	Hg	0.4	29.80	0.085	29.6	30.0
Wind Speed	km/hour	5.0	2.074	0.842	0.2	5.2

After preprocessing data, the time-series data on $PM_{2.5}$ concentration is illustrated in Fig. 2, and descriptive statistics of $PM_{2.5}$ concentration and metrological data are shown in Table 1. To quantify relationship, Pearson's values are calculated and presented in Fig. 3 along with scatter plots. It is observed that $PM_{2.5}$ concentration is correlated with NO_2 the most, followed by pressure, O_3, and humidity. These factors correlate moderately to strongly with $PM_{2.5}$. Precipitation and average temperature have low correlations with $PM_{2.5}$, and wind speed has the lowest correlation.

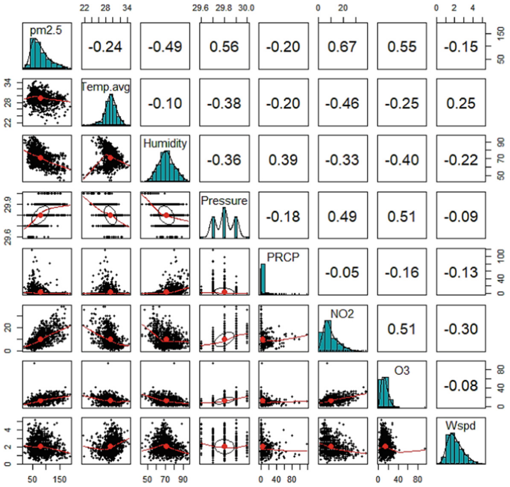

Fig. 3. Scatter plots and Pearson correlations among all pairs of variables

4 Research Methods

4.1 Seasonal ARIMA with Exogenous Covariates

The autoregressive moving average (ARMA) model is a combination of the AR and MA models. The AR model of order p can be written as,

$$\left(1-\phi_1 L-\cdots\phi_p L^p\right)y_t = c+\varepsilon_t, \tag{1}$$

where L^i is a lag operator that converts a variable at time t into its i$^{\text{th}}$-order lagged form, and the MA model of order q is defined as

$$y_t = c+\left(1+\theta_1 L+\cdots+\theta_q L^q\right)\varepsilon_t. \tag{2}$$

The AR component represents the connection between the dependent variable and its previous expression, while the MA term combines the effect of a limited series of random disturbances on the dependent variable. Incorporating differencing and exogenous variables to ARMA model, we obtain a non-seasonal ARIMAX model:

$$\phi(L)\nabla^d y_t = \mathrm{x}_t^T\beta+\theta(\mathrm{L})\varepsilon_{\mathrm{t}}, \tag{3}$$

where $\phi(L)$ is called the autoregressive operator, $\theta(L)$ is called the moving average operator, β is a vector of coefficients, x_t^T is a vector of covariate at time t, and ε_t is a disturbance characterized by a normal distribution with a mean of zero and a constant variance. To describe Eq. (3), notation ARIMAX(p, d, q) is usually used.

The ARIMAX models can also be used to model a variety of seasonal data. Additional seasonal terms are added to the ARIMAX model to create a seasonal ARIMAX model, Eq. (3). It is written as follows:

$$\phi_p(L)\Phi_P(L^S)\nabla^d\nabla^D y_t = \mathrm{x}_t^T\beta + \theta_\mathrm{q}(\mathrm{L})\Theta_\mathrm{Q}(\mathrm{L}^\mathrm{S})\varepsilon_\mathrm{t}, \tag{4}$$

where $\Phi_P(L^S)$ corresponds to a seasonal AR component, $\Theta_Q(L^S)$ corresponds to a seasonal MA component, and S is the duration of the recurring seasonal pattern. The corresponding notation for Eq. (4) is SARIMAX(p, d, q)(P, D, Q)$_S$ [17].

In the R package "forecast", there is a function called auto.arima() that can fit a regression model with ARIMA errors. It employs a variant of the Hyndman-Khandakar method [18] that combines unit root testing, Akaike information criterion (AICc) reduction, and MLE to get the ARIMA model.

4.2 Prophet Model

Facebook [19] created the Prophet model to forecast daily data with weekly and yearly seasonality, as well as holiday influences. It was later extended to incorporate other seasonal data sources. It is effective with time series with strong seasonality and data from many seasons. Prophet is a nonlinear regression model of the following form:

$$y_t = g(t) + s(t) + h(t) + \varepsilon_t, \tag{5}$$

where $g(t)$ represents a piecewise-linear trend, $s(t)$ denotes the various seasonal patterns, $h(t)$ determines the holiday effects, and ε_t is a random error term.

4.3 Regression Tree

Since Breiman [20] proposed decision trees in 1984, statistical learning approaches based on them have grown in popularity. A binary regression tree T divides the space X into many regions as there are leaf nodes, as stated by W. The total prediction function g associated with the tree may be represented as

$$g(\mathrm{x}) = \sum_{w\in W} g^w(\mathrm{x})\mathrm{I}(\mathrm{x} \in \mathbb{R}_W), \tag{6}$$

where I represents the indicator function and $\mathbb{R}_W$ is the region built in the regression tree using logical criteria. The goal of building a tree using a training set $\tau = \{(\mathrm{x}_i, y_i)\}_{i=1}^n$ is to minimize the training squared-error loss,

$$l_\tau(g) = \frac{1}{n}\sum_{w\in W}\sum_{i=1}^{n}\mathrm{I}(\mathrm{x}_i \in \mathbb{R}_W)\left[y_i - g(\mathrm{x}_i)\right]^2. \tag{7}$$

Cost-Complexity Pruning. Let $\tau = \{(\mathrm{x}_i, y_i)\}_{i=1}^{n}$ be a data set and $\gamma \geq 0$ be a real number. For a given tree T, the cost-complexity measure $C_\tau(\gamma, \mathrm{T})$ is defined as:

$$C_\tau(\gamma, \mathrm{T}) = \frac{1}{n} \sum_{w \in W} \sum_{i=1}^{n} \mathrm{I}(\mathrm{x}_i \in \mathbb{R}_W)\big[y_i - g(\mathrm{x}_i)\big]^2 + \gamma |W|, \tag{8}$$

where W denotes the set of terminal nodes of T and $|W|$ denotes the total number of leaves on the tree, which provides insight into its intricacy.

Bootstrap Aggregation. One of the ensemble methods is bootstrap aggregation, commonly known as bagging. There are bootstrap samples $\mathfrak{I}_1^*,\ \mathfrak{I}_2^*,\ \ldots,\ \mathfrak{I}_B^*$ and the matching B independent models giving learner $g_{\mathfrak{I}_1^*},\ g_{\mathfrak{I}_2^*},\ \ldots,\ g_{\mathfrak{I}_B^*}$ from the training set $\mathfrak{I}$ with n observations. The bootstrapped aggregated estimator or bagged estimator is obtained by model averaging as follows:

$$g_{bag}(\mathrm{x}) = \frac{1}{B} \sum_{b=1}^{B} g_{\mathfrak{I}_b^*}(\mathrm{x}). \tag{9}$$

In an idealized situation, the average prediction function converges to the expectation prediction function if $B \to \infty$ and $\mathfrak{I}_1, \mathfrak{I}_2\ ,\ \ldots,\ \mathfrak{I}_B$ are identically and distributed. However, $\mathfrak{I}_1,\ \mathfrak{I}_2,\ \ldots,\ \mathfrak{I}_B$ are not independent, and for large n, the bootstrap sample $\mathfrak{I}^*$ only contains roughly 0.37 of the points from $\mathfrak{I}$[21].

Random Forest. Suppose there is a feature that gives a very excellent split of the data, it will be chosen and divided for every $\{g_{\mathfrak{I}_b^*}\}_{b=1}^{B}$ at the root level, and predictions will be highly correlated. Prediction averaging is unlikely to improve in such a case. this problem is addressed by selecting $m \leq p$ features at random and then calculating splitting criteria. Strong predictors have a lower chance of being retained at the root levels [21].

Conditional Inference Forest. Torsten Hothorn et al. [22] created conditional inference forests (Cforest) to identify the conditional distribution of statistics that quantify the relationships between the response variable and the predictor factors. The Chi-square test statistics are used to examine if any predictors have statistically significant correlations with the response. A global null hypothesis is defined as $H_0 : \bigcap_{j=1}^{m} H_0^j$, where H_0^j indicates that Y is independent of $X_j, j \in \{1, 2,\ \ldots, p\}$.

Gradient Boosted Regression Tree. Any learning algorithm may benefit from boosting, especially if the learner is a poor one. Boosting and bagging both use prediction functions, however the two techniques are fundamentally distinct from each other. Bootstrapped data are used in bagging, while in boosting, the prediction functions are learned in sequentially. At each stage of the boosting round $b,\ b = 1, 2,\ \ldots, B$, a negative gradient on n training points $\mathrm{x}_1, \ldots,\ \mathrm{x}_n$ will be calculated. Next, the negative gradient is estimated using a simple tree by solving

$$h_b = \underset{h \in H}{\arg\min} \frac{1}{n} \sum_{i=0}^{n} \left(r_i^{(b)} - \big[g_{b-1}(\mathrm{x}_i) + h(\mathrm{x}_i)\big]\right)^2. \tag{10}$$

The algorithm makes a γ-sized step in the direction of the negative gradient:

$$g_b(\mathrm{x}) \quad \leftarrow \quad g_{b-1}(\mathrm{x}) + \gamma h_b(\mathrm{x}). \tag{11}$$

Approximation tree learning with sparse data was proposed by Chen and Guestrin [23]. They explain how to build a scalable tree boosting system using caching, compression, and sharing techniques. The combination of these findings allows XGBoost to handle billions of instances while using a fraction of the resources.

Bayesian Additive Regression Tree (BART). The BART model is comprised of a sum-of-trees model plus a regularization prior on model parameters. Let $M = \{\mu_1, \mu_2, \ \ldots, \ \mu_b\}$ denote a set of parameter values for each terminal node b in T and Function $f(x; \mathrm{T}, M)$ that assigns a $\mu_i \in M$ to a single component in $x = (x_1, \ x_2, \ \ldots, \ x_p)$ as follows:

$$Y = \sum_{j=1}^{m} g(x; \mathrm{T}_j, M_j) + \varepsilon, \tag{12}$$

where $\varepsilon \ N(0, \sigma^2)$ and $g(x; \mathrm{T}_j, M_j)$ is the function which assigns $\mu_{ij} \in M_j$ to x. Also, a prior $g(\mathrm{T}_1, M_1), \ \ldots, \ g(\mathrm{T}_m, M_m)$ and σ must be imposed over all sum-of-trees parameters. BART draws posterior samples using MCMC. Chipman et al. [24] describe in detail an iterative Bayesian backfitting MCMC algorithm.

4.4 Support Vector Regression

Vapnik et al. [25] proposed an SVM for regression. Here, $F(\mathrm{x}, \mathrm{w})$ denotes a family of functions parameterized by w, $G(\mathrm{x})$ is an unknown function, and $\hat{\mathrm{w}}$ is the value of w that minimizes an error between $G(\mathrm{x})$ and $F(\mathrm{x}, \hat{\mathrm{w}})$. The representation of $F(\boldsymbol{x}, \hat{\boldsymbol{w}})$ can be defined as

$$F(\mathrm{x}, \hat{\mathrm{w}}) = \sum_{i=1}^{n} \left(\alpha_i^* - \alpha_i\right)\left(\mathrm{v}_i^T \mathrm{x} + 1\right)^p + b, \tag{13}$$

where there are $2n + 1$ values of α_i^*, α_i, and b. The optimum values for the components of $\hat{\mathrm{w}}$ or α depend on a definition of a loss function and the objective function.

4.5 Artificial Neural Network

The artificial neural network (ANN) approach resembles the functioning of human bran, and the algorithm has been based on function:

$$g^*(\boldsymbol{x}) = \sum_{j=1}^{2p+1} h_j\left(\sum_{i=1}^{p} h_{ij}(x_i)\right), \tag{14}$$

where each of the p parts of the input x is expressed as a node in an input layer; there are $2p + 1$ nodes in the hidden layer. The output of a feed-forward neural network with $L + 1$ layers may be expressed as the function composition:

$$g(\mathrm{x}) = S_L \circ M_L \circ \cdots \circ S_2 \circ M_2 \circ S_1 \circ M_1, \tag{15}$$

where $M_l = \mathrm{W}_l z + \mathrm{b}_l,\ l = 1, 2,\ \ldots, L - 1$, S_l is an activation function,W_l is a weight matrix, and b_l is a bias vector.

4.6 K-Nearest Neighbors (KNN) Regression

Let $\tau = \{(\mathrm{x}_i, y_i)\}_{i=1}^{n}$ be a training set and $\{(\mathrm{x}_{(i)}, y_{(i)})\}_{i=1}^{n}$ be a reordering of the data according to increasing distances $\|\mathrm{x}_i - \mathrm{x}\|$ of the $\mathrm{x}_i' s$ to $\boldsymbol{x}$. The usual k-NN regression estimate takes the form $g_n(\mathrm{x}) = \sum_{i=1}^{n} y_{(i)}(\mathrm{x})/k_n$.

5 Results and Conclusions

A total of 1004 data points were used for training and a further 92 for testing. The RMSE, MAE, and Pearson correlation coefficient (PCC) were used to evaluate the forecasts provided by the machine learning models.

Table 2 summarizes and presents the predictive performance indicators for all the 24 models. Based on the training data, the RF has the lowest RMSE and MAE, followed by SARIMAX and trees without pruning; the RF is clearly superior, as its RMSE is only 9.68 and its PCC is close to one. For test data, GBMs with Gaussian and Laplace have the lowest RMSE and MAE, respectively. Based on the PCC, the best three approaches are, respectively, Prophet, NNAR, and GBM with Gaussian distribution.

Considering all the criteria in both the training and test datasets, SARIMAX, trees without pruning, and typical neural networks (except NNAR and ANN using model averaging) tend to be overfitted models, so they are not suitable for $\mathrm{PM}_{2.5}$ prediction. The other models are considered "good" and some of them are evaluated for a particular period, as shown in Fig. 4.

To provide superior forecasts, ensemble techniques employ a collection of machine learning methods. There are numerous "great" models here based on a certain criterion for both training and test datasets. For example, GBM with Gaussian, NNAR, SVR (Poly deg. of 2), BRT, and RF all give RMSE values of less than 20, MAE values of less than 16, and PCC values of greater than 0.7 for the test data. These models shown in Fig. 4 are among the top ten and were included in the ensemble models. There are three kinds of ensemble models: (1) average, (2) median, and (3) weighted. The weights (W) are allocated based on predictive performance: $\mathrm{W}_{\mathrm{GBM}} = 5$, $\mathrm{W}_{\mathrm{NNAR}} = 4$, $\mathrm{W}_{\mathrm{SVR}} = 3$, $\mathrm{W}_{\mathrm{BRT}} = 2$, and $\mathrm{W}_{\mathrm{RF}} = 1$. The predictive performance is shown in Table 3. When compared to all standalone algorithms, the ensemble (weighted) model gives the lowest RMSE, the lowest MAE, and the highest PCC. The ensemble model produced in this work is a mix of the "great" models, which might explain why it performs better.

Table 2. Predictive performance of the univariate and multivariate models

Models		Training data			Test data		
		RMSE	MAE	PCC	RMSE	MAE	PCC
1	SARIMAX (3,1,2)(2,0,0) [7]	13.2309	9.8416	0.9135	29.1229	21.0273	0.3017
2	Prophet model	16.4195	12.4349	0.8625	16.8109	12.9635	**0.7852**
3	Boosted GAM	19.6704	14.8718	0.7958	17.3149	13.3911	0.7487
4	Tree without pruning	13.7452	10.0344	0.9059	27.5612	21.2349	0.5096
5	Tree with pruning	19.7211	14.9438	0.7942	19.7729	14.9105	0.7017
6	Bagged regression TREE	16.5598	12.8007	0.8644	17.9855	13.8474	0.7280
7	Random forest (RF)	**9.6836**	**7.1578**	**0.9617**	18.8680	14.7794	0.7065
8	Conditional RF	16.5503	12.2456	0.8626	18.1553	13.8912	0.7169
9	GBM with Gaussian	18.2094	13.6183	0.8288	**16.7995**	12.8349	0.7603
10	GBM with Student-t	20.1617	14.2879	0.7971	17.5609	12.8646	0.7345
11	GBM with Laplace	19.3167	13.5126	0.8122	16.9995	**12.5213**	0.7578
12	XGBoost	16.2301	12.1305	0.8686	17.3822	13.1277	0.7448
13	BART	17.9175	13.5989	0.8346	17.7948	13.5950	0.7357
14	7-11-1 ANN (logistic)	17.4170	13.0207	0.8438	23.6783	19.4137	0.6601
15	7-11-1 ANN (tanh)	17.5802	13.3950	0.8406	31.6012	26.3459	0.2858
16	7-11-4-1 ANN (logistic)	15.9349	11.8504	0.8712	25.7745	21.3557	0.5825
17	7-11-4-1 ANN (tanh)	14.8501	11.1096	0.8892	53.2947	40.3602	0.3647
18	NNAR (29,1,18)[7]	13.5594	9.0555	0.9095	17.3663	12.9768	0.7772
19	ANN using model averaging	19.8413	15.0865	0.7950	18.5319	14.5525	0.7132
20	KNN (Rectangular)	18.4564	13.6578	0.8288	19.5365	15.7029	0.6755

(*continued*)

Table 2. (*continued*)

Models		Training data			Test data		
		RMSE	MAE	PCC	RMSE	MAE	PCC
21	KNN (Epanechnikov)	19.2517	14.2518	0.8103	19.4960	15.6700	0.6782
22	SVR Linear	21.6400	16.2817	0.7453	17.0003	13.1713	0.7562
23	SVR (Poly deg. of 2)	20.2533	14.5404	0.7848	17.9685	13.9103	0.7277
24	SVR (Poly deg. of 3)	19.2560	13.4661	0.8075	21.9772	16.3793	0.6050

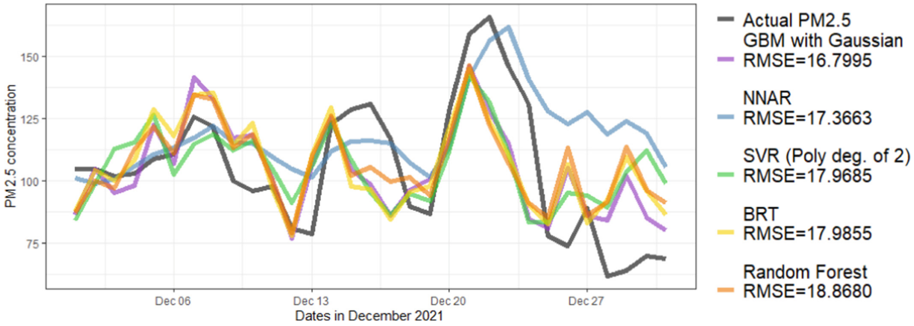

Fig. 4. Forecasts from the selected models compared to the actual values of $PM_{2.5}$

Table 3. Predictive performance on test data for the ensemble models

Model	RMSE	MAE	PCC
Ensemble (mean)	16.3746	12.6136	0.7763
Ensemble (median)	17.1361	13.0713	0.7501
Ensemble (weighted)	15.9516	12.3144	0.7888

References

1. Jung, R., Hwang, F., Chen, T.: Incorporating long-term satellite-based aerosol optical depth, localized land use data, and meteorological variables to estimate ground-level PM 2.5 concentrations in Taiwan from 2005 to 2015. Environ. Pollut. **237**(1), 1000–1010 (2018)
2. Health Effects Institute: State of Global Air 2020. Special Report. Boston, MA (2020)
3. Yiyi, W., et al.: Associations of daily mortality with short-term exposure to PM2.5 and its constituents in Shanghai, China. Chemosphere **233**, 879–887 (2019)
4. Xing, Y.F., et al.: The impact of PM2.5 on the human respiratory system. J. Thorac. Dis. **8**(1), E69–E74 (2016)
5. World Air Quality Index project. https://aqicn.org/city/bangkok/. Last accessed 25 Mar 2022

6. Catalano, M., et al.: Improving the prediction of air pollution peak episodes generated by urban transport networks. Environ. Sci. Policy. **60**, 69–83 (2016)
7. Masood, A., Ahmad, K.: A model for particulate for Delhi based on machine learning approaches. Procedia. Comput. Sci. **167**, 2101–2110 (2020)
8. Suleiman, A., et al.: Applying machine learning methods in managing urban concentrations of traffic-related particulate matter (PM_{10} and $PM_{2.5}$). Atmos. Pollut. Res. **10**(1), 134–144 (2019)
9. Doreswamy, et al.: Forecasting air pollution particulate matter ($PM_{2.5}$) using machine learning regression models. Procedia. Comput. Sci. **171**, 2057–2066 (2020)
10. Sharma, N., et al.: Forecasting air pollution load in Delhi using data analysis tools. Procedia. Comput. Sci. **132**, 1077–1085 (2018)
11. Qiao, W., et al.: The forecasting of $PM_{2.5}$ using a hybrid model based on wavelet transform and an improved deep learning algorithm. IEEE Access **7**, 142814–142825 (2019)
12. Biancofiore, F.: Recursive neural network model for analysis and forecast of PM_{10} and $PM_{2.5}$. Atmos. Pollut. Res. **8**, 652–659 (2017)
13. Mahajan, S.: An empirical study of $PM_{2.5}$ forecasting using neural network. In: 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation, pp. 1–7. IEEE, San Francisco, USA (2017)
14. Ejohwomu, O.A., et al.: Modelling and forecasting temporal $PM_{2.5}$ concentration using ensemble machine learning methods. Buildings **12**(1), 46 (2022)
15. Gupta, P., et al.: Machine learning algorithm for estimating surface $PM_{2.5}$ in Thailand, Aerosol Air Qual. Res. **21**(11), 210105 (2021)
16. Buuren, S.: Karin Groothuis-Oudshoorn: mice: multivariate imputation by chained equations in R. J. Stat. Softw. **45**(3), 1–67 (2011)
17. Box, G.E.P., et al.: Time series analysis: forecasting and control, 4th edn. John Wiley & Sons Inc., Hoboken, New Jersey (2008)
18. Hyndman, R.J., Khandakar, Y.: Automatic time series forecasting: the forecast package for R. J. Stat. Softw. **27**(3), 1–22 (2008)
19. Taylor, S.J., Letham, B.: Forecasting at scale. Am. Stat. **72**(1), 37–45 (2018)
20. Breiman, L., et al.: Classification and Regression Trees, 1st edn. Chapman and Hall/CRC, Boca Raton (1984)
21. Kroese, D.P., et al.: Data Science and Machine Learning: Mathematical and Statistical Methods, 1st edn. Chapman and Hall/CRC, Boca Raton (2020)
22. Hothorn, T., et al.: Unbiased recursive partitioning: a conditional inference framework. J. Comput. Graph. Stat. **15**(3), 651–674 (2006)
23. Chen, T.Q., Guestrin, C.: XGBoost: a scalable tree boosting system. https://arxiv.org/abs/1603.02754. Last accessed 17 Apr 2022
24. Chipman, H.A., et al.: BART: Bayesian additive regression trees. Ann. Appl. Stat. **4**(1), 266–298 (2010)
25. Vapnik, V., et al.: Support vector method for function approximation, regression estimation, and signal processing. Adv. Neural Inf. Process. Syst. **9**, 281–287 (1997)