Air pollution forecasting: Application of machine learning models to estimate PM2.5 index

Nghiem Van Tinh*, Pham Quang Hieu

Faculty of Electronics - Thai Nguyen University of Technology, No. 666 3/2 Road, Thai Nguyen City, Thai Nguyen, Vietnam.
*Corresponding author: nghiemvantinh@tnut.edu.vn
Received 27 Aug. 2024; Revised 25 Oct. 2024; Accepted 15 Nov. 2024; Published 6 Dec. 2024.
DOI: https://doi.org/10.54939/1859-1043.j.mst.FEE.2024.286-294

ABSTRACT

In the context of ongoing industrialization, air pollution has become an urgent global problem, particularly severe in large cities such as Hanoi (Vietnam), Beijing (China), and others. Air pollution, especially the concentration of fine particulate matter (PM2.5), is not only harmful to human health but also has significant negative impacts on the environment, economy, and quality of life. This study aims to enhance the ability to predict air pollution levels more accurately. By using machine learning models, meteorologists can better predict air pollution levels and propose more effective mitigation solutions. The article utilizes a multivariate time series dataset, including meteorological and air pollution indices from Beijing, China, from 2010 to 2014. Machine learning models such as Lasso Regression, Support Vector Regression, Random Forest, XGBoost, and, notably, a Stack Model combining the four aforementioned models, are evaluated. The performance of these models is measured using statistical indicators such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Coefficient of Determination (R^2). Among these models, the Stack model provides the most accurate predictions for the PM2.5 index.

Keywords: Air quality prediction; Machine learning; Random forest; SVR; XGBoost; PM2.5.

1. INTRODUCTION

Air pollution is increasingly causing serious health issues. According to the World Health Organization (WHO), it is a global problem affecting everyone [1]. A major contributor is the rising concentration of PM2.5 particles, which, due to their small size, can easily enter the lungs and cause severe health problems, even death [2]. PM2.5 data is collected from environmental monitoring stations, but uneven distribution of these stations leads to data discrepancies across regions. Portable PM2.5 monitoring devices and satellite imagery offer potential solutions to fill these data gaps, especially in areas lacking high-precision stations [3, 4]. PM2.5 pollution also affects certain groups differently, as shown by He et al. [5], who found a correlation between PM2.5 levels and increased acute myocardial infarction rates in women during emergency cases. Moreover, 98% of low-and middle-income countries with populations over 100,000 do not meet WHO's air quality standards for PM2.5 levels.

In recent years, machine learning methods have demonstrated their ability to capture nonlinear relationships and are increasingly applied in air pollution forecasting. Machine learning models, such as neural networks, regression methods, and reinforcement learning, have shown superior performance in predicting PM2.5 levels [6, 7]. Recent studies also indicate that machine learning outperforms traditional statistical methods in identifying relationships between variables and improving forecast accuracy [8, 9]. Additionally, machine learning algorithms often require less historical data than traditional methods to achieve similar accuracy in identifying relationships between explanatory variables and the target variable [10]. With the advent of machine learning, time series models can better capture dynamic relationships between variables over time, providing deeper insights into past trends. In this study, we use five machine learning models to estimate PM2.5 concentrations over a specific period. The dataset is split into two parts: 90% for training

and 10% for testing. Training data is used to train the models, while testing data is used to evaluate the performance of the trained models.

2. MACHINE LEARNING MODELS

This section briefly presents the machine learning models applied to time series forecasting in the study as follows.

2.1. Lasso regression (LR)

Lasso Regression [11] is a linear regression method that uses the L1 norm to regularize the model. This prevents the model from overfitting and has the ability to eliminate some unnecessary features by setting their coefficients to zero.

2.2. Support Vector Regression (SVR)

Support Vector Regression (SVR) [12] is a regression algorithm within the Support Vector Machine family, designed to find a hyperplane that best approximates the relationship between input and target variables. This hyperplane is defined by support vectors, which are key data points influencing its position and orientation [13]. SVR, particularly useful for time series prediction, gained attention for its ability to create nonlinear boundaries using kernel functions [14].

2.3. Random Forest (RF)

RF [15] is an ensemble learning method that builds multiple decision trees during training and outputs the average prediction of each tree. This approach improves prediction accuracy and controls overfitting by averaging multiple models trained on different subsets of data. It has good noise reduction capabilities and can handle datasets with multiple input variables.

2.4. XGBoost (Extreme Gradient Boosting)

XGBoost [16] is an optimized gradient boosting algorithm used for both classification and regression tasks [17]. It employs multiple decision trees as weak learners, trained via gradient descent to minimize the loss function. XGBoost leverages parallelization, distributed computing, and out-of-memory processing, making it a powerful and efficient tool for handling large and complex datasets in regression tasks.

2.5. Stacking (stacked generalization)

Stacking [18] is an ensemble method that improves predictive performance by merging multiple base models. Instead of simply averaging or relying on the majority rule like bagging or boosting, stacking uses another machine learning model (called meta-model) to optimize the combination of predictions from the base models. Figure 1 illustrates the ensemble process of the Stacking model.

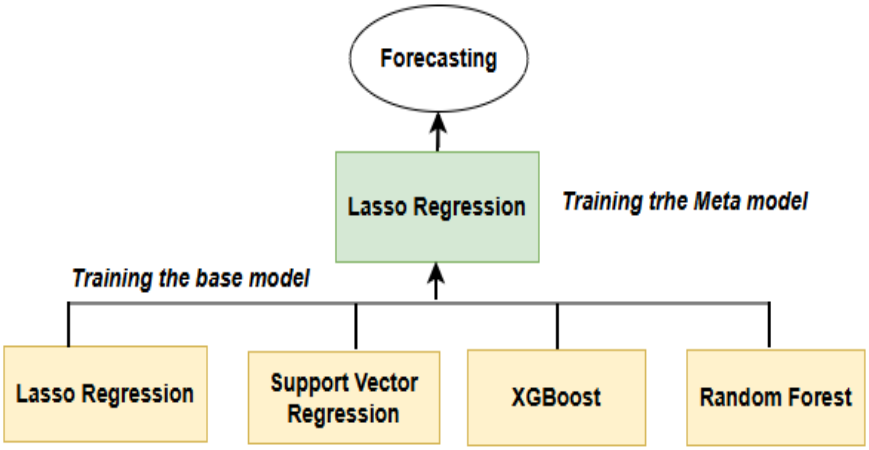

Figure 1. *Illustration of the Stacking learning model.*

3. APPLYING MODELS AND FORECASTING

Applying machine learning models to time series forecasting involves several steps, as summarized in figure 2 below:

3.1. Data requirements

3.1.1. Data Collection

The data, collected from air pollution and meteorological records in Beijing, China (2010-2014) [19], is available in the UCI Machine Learning Repository at https://archive.ics.uci.edu/dataset/381/beijing+pm2+5+data. It includes hourly data on weather conditions such as dew point, temperature (°C), pressure (hPa), wind direction, cumulative wind speed (m/s), and PM2.5 concentration (µg/m³). The dataset's structure is summarized in table 1 as follows:

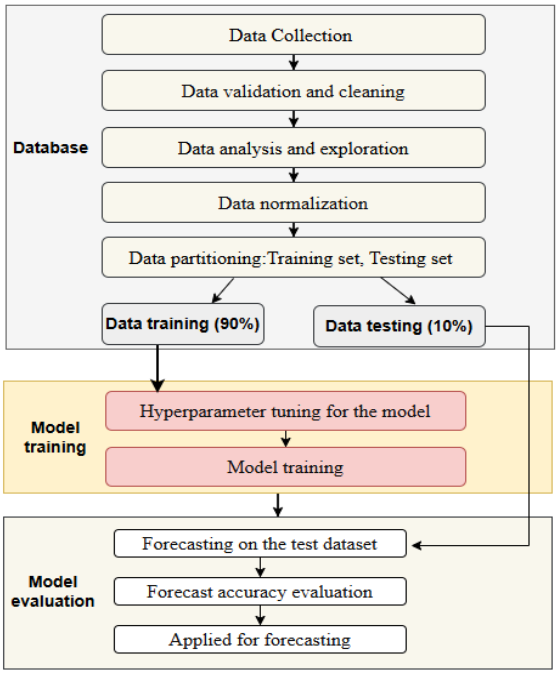

Figure 2. *Steps to build and evaluate the forecasting model.*

Table 1. *Information about the columns in the original dataset.*

Column name	Data type	Meaning
no	*int64*	*The sequence number of the record in the data set*
year	*int64*	*Year data recorded (from 2010-2014)*
month	*int64*	*Month when data was recorded (January-December)*
pm2.5	*float64*	*Is the concentration of fine dust particles with a diameter of 2.5 micrometers or less, measured in (µg/m³).*
...	...	.....

3.1.2. Data validation and cleaning

The current dataset consists of 13 data fields with a total of 43824 rows of records. All missing data in the PM2.5 column were replaced with zero values, and the columns recording the time were discarded. In addition, the first 24 hours of the dataset were also discarded. When applying different methods, the dataset was converted into a time series so that it could be used to solve the supervised learning problem [20]. To clean the data and make it easier to visualize the variables, some of the variables in the columns of table 1 are transformed into variables in table 2: for example, "is" becomes "cs"; "ir" becomes "cr"; and so on.

Table 2. *Statistical index of columns with data types int64, float64.*

	hour	pm2.5	dewp	temp	pres	cws	cs	cr
count	43824	41757	43824	43824	43824	43824	43824	43824
mean	11,50	98,61	1,82	12,45	1016,45	23,89	0,05	0,19
sdt	6,92	92,05	14,43	12,20	10,27	50,01	0,76	1,42
min	0,00	0,00	-40,00	-19,00	991,00	0,45	0,00	0,00
25%	5,75	29,00	-10,00	2,00	1008,00	1,79	0,00	0,00
50%	11,50	72,00	2,00	14,00	1016,00	5,37	0,00	0,00
75%	17,25	137,00	15,00	23,00	1025,00	21,91	0,00	0,00
max	23,00	994,00	28,00	42,00	1046,00	585,60	27,00	36,00

After inspection, it was found that 41,757 records (about 5% of the total 2,067 records) in the "PM2.5" column were erroneous. Specifically, 669 values were missing in 2010, and 728 in 2011. Nearly 5% of the PM2.5 data contained NaNs, with two-thirds concentrated in 2010 and 2011. This is problematic as most training data comes from this period, and missing or erroneous values could affect model performance. To address this, we will apply a three-step process:

- Remove observations with missing values on January 1, 2010.
- Interpolate missing values by hour of the day (from 0:00-14:00 and 15:00-23:00).
- Replace any remaining missing values with the median value.

3.1.3. Data analysis and exploration

In fact, when analyzing hourly forecast data, the chart provides only an overview of the target value. However, it also clearly shows the complexity of seasonality and the strong fluctuations in "PM2.5" concentrations on an hourly basis. High values of "PM2.5" indicate severe air pollution levels during certain periods. To better understand the dispersion of data by year, a box plot can be used, as shown in figure 3.

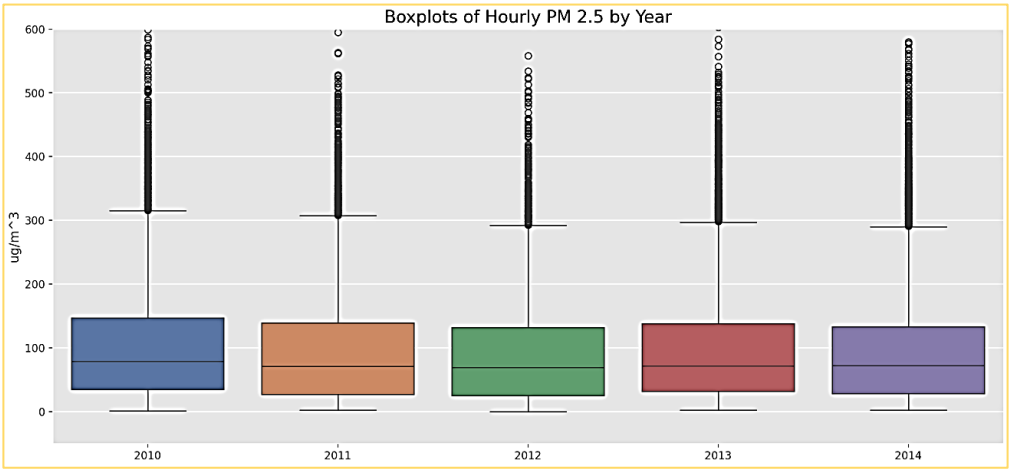

Figure 3. *Box plot depicting the data distribution of "pm25" values over the years.*

Figure 3 shows that the PM2.5 data for each year has a fairly consistent distribution, suggesting stable pollution levels across the years. However, outliers are present in all years, particularly in 2010, 2013, and 2014, which show more frequent spikes in PM2.5 levels compared to 2011 and 2012. Next, we removed unnecessary variables and added new ones by evaluating their correlation with PM2.5. Based on the correlation matrix in figure 4, we created a "target" variable by shifting the "PM2.5" values up by one row. Weather variables like "dewp", "temp", and "pres" showed weak correlation with the target, with "pres" being the least correlated. To reduce multicollinearity and improve model performance, we removed "pres", "cr", and "cs".

Figure 4. *Correlation matrix between variables in the dataset.*

3.1.4. Normalize and split the data set

This study uses Python's scikit-learn StandardScaler to standardize the input data by removing the mean and dividing by the standard deviation, resulting in a mean of 0 and a standard deviation of 1. The tool is applied to standardize three datasets: "X_base" (base model training data), "X_meta" (meta model training data), and "X_valid" (test data).

Initially, with 100% data correspondence: 43797 rows. Divided into two stages:

Phase 1: The initial 90% of the data is further divided into two parts:

- Take the first 2/3 of the data to train the 4 base models.
- Take the remaining 1/3 of the data to train the Meta model.

Phase 2: The last 10% of the data (hold-out) in the dataset is used as the final test dataset.

3.2. Model training

In this study, we select the hyperparameters for each base model using the "GridSearchCV" technique. The hyperparameters corresponding to each model are reported in table 3.

Table 3. *Hyperparameters selected for each model.*

Model	Hyperparameters	Search domain	Hyperparameter values
Lasso regression	fit_intercept	True, False	True
	alpha	0.005, 0.01, 0.03, 0.05, 0.07, 0.1	0.05
SVR	epsilon	8, 9	9
	fit_intercept	True, False	True
	C	33, 34	34
Random forest	n_estimators	range(300, 500, 25)	375
	min_samples_split	2, 3, 4, 5, 6, 7	2
	max_features	log2, sqrt	sqrt
XGBoost	n_estimators	range(70, 140, 10)	120
	subsample	0.5, 0.75, 1	0.75
	Eta	0.01, 0.05, 0.1, 0.2, 0.3, 0.4	0.05
	gamma	range(150, 310, 10)	260

Model training process

Step 1: Train the base models on the first 2/3 of the 90% of the data.

Step 2: Use the base models as the parameter optimization models above to predict the remaining 1/3 of the 90% of the data.

Step 3: Create a new dataset from these predictions, with the structure:

- Input variable: combine the predicted values of the four base models with the remaining 1/3 of the data.
- Target variable: Actual value from the original 1/3 of the data.

Step 4: Train the meta model (LR) on this new dataset.

Model testing process

Step 1: Pass the test dataset through the base models.

Step 2: Create a new test dataset from these predicted values, with the structure:

- Input variable: match the predicted values of the four base models with the test dataset;
- Target variable: the actual output value of the test set.

Step 3: Test the meta model (LR) on this new dataset.

Step 4: Calculate the values of evaluation criteria: R^2, RMSE, MAE and conclusion.

Performance evaluation criteria of models

The model's performance is evaluated using statistical criteria such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and coefficient of determination (R^2), calculated using the following formulas:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|F_i - R_i| \quad (2)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(F_i - R_i)^2} \quad (3)$$

$$R^2 = 1 - \frac{\sum_i^n (F_i - R_i)^2}{\sum_i^n (F_i - \overline{R})^2} \quad (4)$$

In which, F_i is the forecast value at time i, $\overline{R}$ is the actual value at time i, n is the total number of sample data participating in the forecast, $\overline{R}$ is the average value of the actual data.

3.3. Model prediction and evaluation results

The results of the models were implemented using Python 3.12.5 and the Scikit-Learn library. Figure 5 shows that most models accurately predict the data trends, with predicted values close to actual values. However, larger errors occur when predicting peak or trough values, with models underestimating peaks and overestimating troughs. This issue is most prominent in the Random Forest model, while the Stack model demonstrates significantly smaller errors.

Next, we perform a comparison between the forecasting models based on the RMSE, MAE, and R^2 criteria, as shown in table 4. The results in table 4 show that the Stack model has the best performance compared to the other models. However, the individual models also achieve quite high accuracy, especially the XGBoost model.

Table 4. *Comparison of forecast errors between models.*

Models	MAE	RMSE	R^2
Lasso Regression	10.9421	18.8500	0.9551
SVR	11.6184	19.1148	0.9539
Random Forest	12.3886	20.5072	0.9469
XGBoost	10.7346	18.8468	0.9551
Stack Model	10.7072	18.7581	0.9556

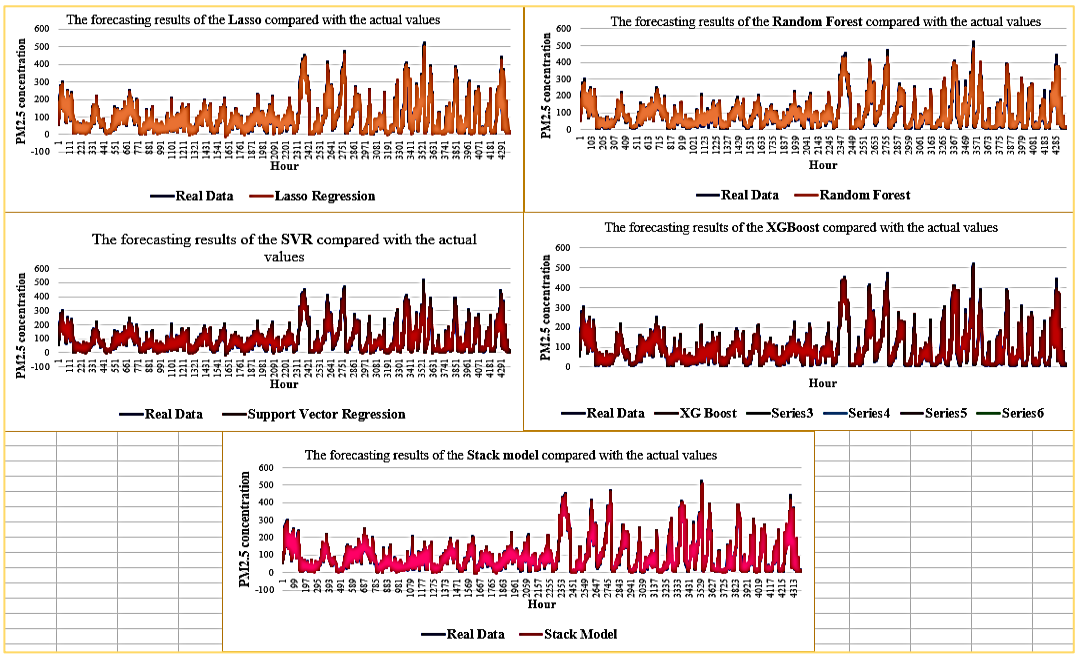

Figure 5. *Graph showing the predicted values of each model compared to the actual values.*

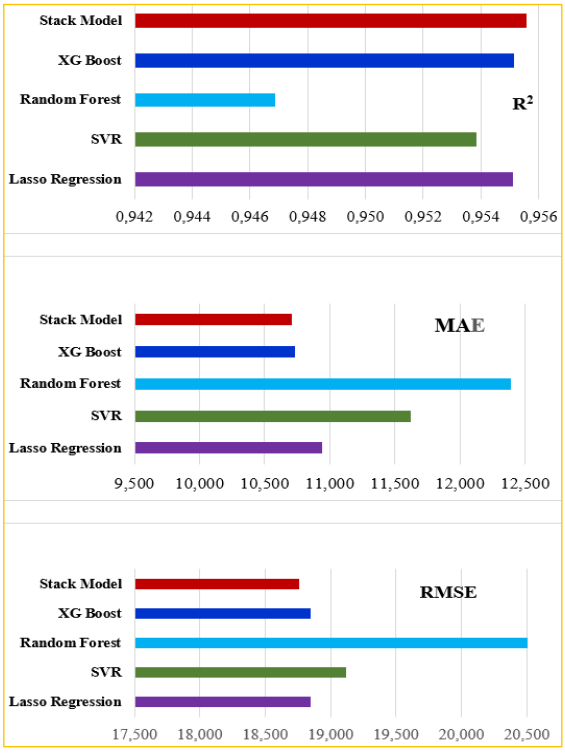

Figure 6. *Comparison of model forecast errors based on R^2, MAE, RMSE criteria.*

More intuitively, the forecast errors of the models are shown in figure 6. The comparison results show that the XGBoost model and the Stack model have the highest accuracy, with R^2 of approximately 0.956 and the smallest error with MAE of approximately 10.7 and RMSE of approximately 18.76. Then, the Lasso Regression model and finally the Random Forest model give worse results.

Figure 7 compares the model predictions with the actual value at a PM2.5 peak. The results show that, although no model predicted the peak value exactly, most of the models predicted values quite close to the actual value. In particular, the Stack model predicted the closest to the actual value, showing the potential for improving the error when forecasting peak values.

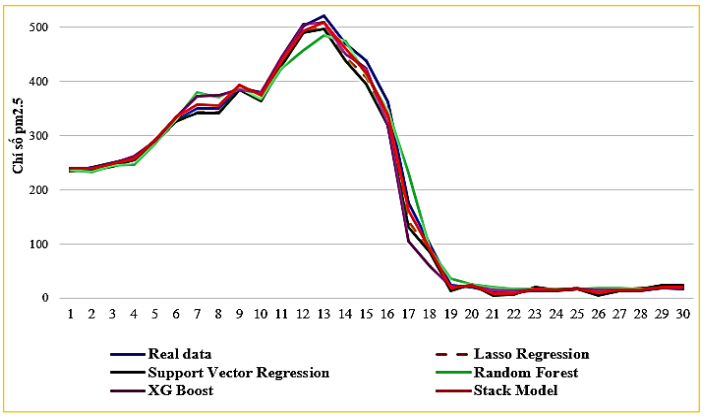

Figure 7. *Comparison of predicted values of models with actual values based on peak values.*

4. CONCLUSIONS

The conclusion of this study highlights the goal of improving PM2.5 forecasting by comparing five machine learning models: Lasso Regression, Support Vector Regression, Random Forest, XGBoost, and the Stack Model. The Stack Model outperforms the others, delivering the most accurate forecasts with the lowest RMSE and highest R^2, thanks to its ability to combine the strengths of individual models. While XGBoost also performs well, it slightly lags behind the Stack Model. Lasso Regression and SVR show good accuracy, while Random Forest performs the worst. In the face of growing air pollution in major cities, this research provides a more effective forecasting tool and sets the stage for advancing prediction models, especially for use in other regions of Vietnam, ultimately aiding in air pollution reduction and public health protection.

REFERENCES

[1]. WHO, *"Air pollution,"* (2018). [Online]. Available: https://www.who.int/en/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health.

[2]. Wei, J., Huang, W., Li, Z., Xue, W., Peng, Y., Sun, L., & Gribb, M. *"Estimating 1-km-resolution PM2.5 concentrations across China using the spacetime random forest approach"*. Remote Sensing of Environment, 231, 111221, (2019). https://doi.org/10.1016/j.rse.2019.111221

[3]. Viana, M., Rivas, I., Reche, C., Fonseca, A., Pérez, N., Querol, X. *"Field comparison of portable and stationary instruments foroutdoor urban air exposure assessments"*. Atmospheric Environment, 123, 220–228, (2015). https://doi.org/10.1016/j.atmosenv.2015.10.076

[4]. Motlagh, N. H., Lagerspetz, E., Nurmi, P., Li, X., Varjonen, S., Mineraud, J. *"Toward massive scale air quality monitoring"*. IEEECommunications Magazine, 58(2), 54–59, (2020). https://doi.org/10.1109/MCOM.001.1900515

[5]. He, X. N., Chen, P., Zhang, C., & Chen, J. Y. *"Study on the correlation between PM 2.5 and onset of acute myocardial infarction amongfemale patients"*. Child Care China, 31(22), 4626–4629, (2016).

[6]. Ong, B. T., Sugiura, K., & Zettsu, K. *"Dynamically pre-trained deep recurrent neural networks using environmental monitoring data forpredicting PM2.5"*. Neural Computing and Applications, 27(6), 1553–1566, (2016). https://doi.org/10.1007/s00521-015-1955-3

[7]. Fang, X., Zou, B., Liu, X., Sternberg, T., & Zhai, I. *"Satellite-based ground PM 2.5 estimation using timely structure adaptive modeling"*. Remote Sensing of Environment, 186, 152–163, (2016). https://doi.org/10.1016/j.rse.2016.08.027

[8]. Bzdok, D., Altman, N., & Krzywinski, M. *"Statistics versus machine learning"*. Nature Methods, 15(4), 233–234, (2018). https://doi.org/10.1038/nmeth.4642

[9]. Bazoukis, G., Stavrakis, S., Zhou, J., Bollepalli, S. C., Tse, G., Zhang, Q. *"Machine learning versus conventional clinical methods in guiding management of heart failure patients - A systematic review"*. Heart Failure Reviews, 26(1), 23–34, (2021). https://doi.org/10.1007/s10741-020-10007-3

[10]. Makridakis, S., Spilliotis, E., & Assimakopoulos, V. *"Statistical and machine learning forecasting methods: Concerns and ways forward"*. PLoS One, 13(3), e0194889, (2018). https://doi.org/10.1371/journal.pone.0194889

[11]. Ranstam, J. and J.A.J.J.o.B.S. Cook, *"LASSO regression"*. 105(10): p. 1348-1348, (2018).

[12]. Awad, M., et al., *"Support vector regression"*. p. 67-80, (2015).

[13]. Hastie, T. J., Tibshirani, R. J., & Friedman, J. H. *"The elements of statistical learning: Data mining inference and prediction (2nd ed.)"*. Springer, (2009).

[14]. Bao, Y., Hayashida, M., & Akutsu, T. *"LBSizeClev: Improved support vector machines (SVM)-based predictions of Dicer cleavage sitesusing loop/bulge length"*. BMC Bioinformatics, 17(1), 487, (2016). https://doi.org/10.1186/s12859-016-1353-6

[15]. Segal, M.R., *"Machine learning benchmarks and random forest regression"*. (2004).

[16]. Chen, T. and C. Guestrin. *"Xgboost: A scalable tree boosting system"*. Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. (2016).

[17]. Chen, T., & Guestrin, C. *"XGBoost: A scalable tree boosting system"*. In Proceedings of the ACM SIGKDD International Conference onKnowledge Discovery and Data Mining (pp. 785–794), (2016). https://doi.org/10.1145/2939672.2939785

[18]. Wolpert, D.H., *"Stacked generalization"*. Neural Networks. 5(2): p. 241-259, (1992).

[19]. X. Liang et al., *"Assessing Beijing's PM 2.5 pollution: severity, weather impact, APEC and winter heating"*. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science, vol. 471, no. 2182, p. 20150257, (2015).

[20]. J. Brownlee, *"How to Convert a Time Series to a Supervised Learning Problem in Python,"* (2017). https://machinelearningmastery.com/convert-time-series-supervised-learning-problem-python/.

TÓM TẮT

Dự báo ô nhiễm không khí: Ứng dụng các mô hình học máy để ước lượng chỉ số PM2.5

Trong bối cảnh công nghiệp hóa hiện nay, ô nhiễm không khí đã trở thành một vấn đề toàn cầu cấp bách, đặc biệt nghiêm trọng tại các đô thị lớn như Hà Nội - Việt Nam, Bắc Kinh - Trung Quốc,... Ô nhiễm không khí, đặc biệt là mật độ hạt bụi mịn PM2.5 (gọi tắt là PM2.5) không chỉ gây hại cho sức khỏe con người mà còn tác động tiêu cực đến môi trường, kinh tế và chất lượng cuộc sống. Do đó, việc sử dụng mô hình học máy để đánh giá mức độ ô nhiễm PM2.5 tại mỗi thành phố nói riêng và trên toàn cầu nói chung là cấp thiết. Điều này giúp các nhà khí tượng học có thêm công cụ để dự đoán chính xác hơn mức độ ô nhiễm không khí và đưa ra các giải pháp giảm thiểu hiệu quả. Bài báo này sử dụng bộ dữ liệu chuỗi thời gian đa biến, bao gồm các chỉ số khí tượng và chỉ số ô nhiễm không khí tại Bắc Kinh, Trung Quốc trong giai đoạn từ năm 2010 đến 2014, để đánh giá dựa trên các mô hình học máy như: Lasso Regression (LR), Support Vector Regression (SVR), Random Forest (RF), XGBoost (XGB), và đặc biệt là mô hình Stack Model kết hợp từ bốn mô hình trên. Hiệu suất của các mô hình này được đo lường bằng các chỉ số thống kê như: Sai số bình phương trung bình (RMSE), sai số tuyệt đối trung bình (MAE) và hệ số xác định (R^2). Trong các mô hình trên, Stack model đưa ra kết quả dự báo chỉ số PM 2.5 tốt nhất.

Từ khóa: Dự báo mức độ không khí; Học máy; Dừng ngẫu nhiên (RF); SVR; XGBoost; PM2.5.