EMITTER International Journal of Engineering Technology
Vol. 13, No. 2, December 2025, pp. 198~213
DOI : 10.24003/emitter.v13i2.933

Performance Analysis of Decision Tree Ensemble Models and Feature Importance Analysis in Prediction of Particulate Matter PM10

Sherin Babu[1,3*], Binu Thomas [2,3]

[1]Department of Computer Science, Assumption College Autonomous, Changanassery, Kottayam, Kerala.
[2]Department of Computer Applications, Marian College, Kuttikanam, Idukki, Kerala.
[3]School of Computer Sciences, Mahatma Gandhi University, Kottayam, Kerala.
Corresponding Author: sherinbabu@assumptioncollege.edu.in

Received August 19, 2025; Revised September 26, 2025; Accepted November 4, 2025

Abstract

Particulate Matter induced air pollution is known to have significant negative impacts on both the environment and human health. This research evaluates the effectiveness of various decision tree ensemble models in predicting daily PM10 concentrations in Thiruvananthapuram, Kerala, from July 2017 to December 2019. Seven decision tree ensemble models, namely Random Forest, Extra Trees, Gradient Boosting, AdaBoost, LightGBM, XGBoost, and Histogram-Based Gradient Boosting are employed here. To address missing data in the dataset, kNN imputation is utilized for a cohesive dataset suitable for model training. The models utilize both meteorological and air pollutant variables, with performance assessment using metrics such as the coefficient of determination (R^2), root mean square error (RMSE) and mean absolute error (MAE). The findings indicate that the Extra Trees regression model provided the best prediction performance (R^2 = 0.9397, RMSE = 6.664 μg/m^3, MAE = 4.950 μg/m^3). Histogram-Based Gradient Boosting and Random Forest also demonstrate strong predictive capabilities. The explainability of the best prediction models is conducted by the feature importance analysis process. Feature importance analysis highlighted sulfur dioxide (SO2) as the most significant pollutant influencing PM10 levels, alongside meteorological factors like wind speed and rainfall, enhancing both prediction accuracy and interpretability of results. This research represents the first comprehensive effort to predict PM10 levels in Thiruvananthapuram using machine learning techniques, addressing a gap in regional air quality studies.

Keywords: PM10, Decision Tree, Extra Trees, Random Forest, Histogram Gradient Boosting.

Copyright © 2025 EMITTER International Journal of Engineering Technology - Published by PENS

1. INTRODUCTION

Particulate matter is the total amount of solid and liquid particles suspended in the air, many of which are harmful. Inhalable particles with a diameter size of 10 micrometres or less are referred to as PM10 (1). Asthma, respiratory infections, lung cancer, and chronic obstructive pulmonary disease(COPD) are all caused by exposure to PM10 (2). The elderly and children with chronic heart or lung disease are most likely to suffer negative health effects from PM10 exposure, according to the researchers (3,4). Increased PM10 concentrations have been linked to a higher mortality rate (5). The prevalence of cardiovascular disease (CVD) in India is one of the highest in the world. The annual number of CVD deaths in India is anticipated to rise from 2.26 million in 1990 to 4.77 million in 2025 (6,7).

Innovations in computational methods and the availability of large amount of data storage devices, have resulted in the development of applications for predicting air pollutant concentrations for a spectrum of uses (8). Machine learning algorithms have been successfully applied to the forecasting of a wide range of air pollutant concentrations over a variety of time scales (9). The accuracy and dependability of air quality forecasts are improved by utilising a variety of machine learning methods for both short-term and long-term PM10 level prediction (10). The prediction of PM10 levels is benefiting greatly from the use of traditional regression techniques, tree-based models, hybrid models, and deep learning techniques, which provide improved accuracy and the capacity to efficiently handle massive amounts of environmental data. These developments have a major impact on environmental management plans and public health campaigns that try to reduce air pollution.

2.RELATED WORKS

Many researchers tried to forecast particulate matter concentrations, both PM10 and PM2.5 using machine learning methods. The researchers developed a neural network based model that accepted PM10 concentrations and meteorological parameters as input variables, in order to predict the PM10 concentration for the next day (11). A random forest model that uses satellite, meteorologic, atmospheric, and land-use data for predicting daily PM2.5 concentrations at a resolution of 1 × 1 km throughout an urban area, was proposed in this study (12). The researchers described a successive over relaxation support vector regress (SOR-SVR) model for the PM10 and PM2.5 prediction, based on the daily average aerosol optical depth (AOD) and meteorological parameters measured in Beijing during the years 2010 to 2012 (13). An ANN-SVM forecasting model with a two-year data set of air pollutant and meteorological parameters from Taiyuan, China, and then the Taylor expansion forecasting model to revise the forecasting goal, resulting in a high accuracy rate is reported in a research study (14). In the research work of (15) , the researchers proposed a PM10 forecast model focused on Long Short Term Memory (LSTM) for Seoul, Korea.

A comparative study of Artificial Neural Networks (ANN), Boosted Regression Trees (BRT), and SVM machine learning models to predict PM10 and PM2.5 levels based on traffic, meteorological, and pollutant data collected from various locations in London from 2007 to 2012 is presented in the study (16). The researchers developed a model to estimate daily concentrations of PM1, PM2.5, PM4, PM10, and PM-Total based on weather variables by employing the hybrid dragonfly-SVM algorithm (17). Fuzzy inference system optimized using particle swarm optimization and genetic algorithm was developed by the researchers for forecasting PM10 and other air pollutants (18). The research work that predicts the hourly concentration of PM10 in Seoul using tree-based machine learning reported that LightGBM model was superior in prediction performance (19). ANN with different network training algorithms was employed to predict hourly PM10 concentrations in Chongqing City of China (20). A hybrid deep learning method of encoder-decoder convolutional neural network combined the Long Short-Term Memory (LSTM) model was developed for PM10 prediction and it reported R^2 value of 0.88 and a mean absolute error value of 7.24 (21).

3. ORIGINALITY

In this study, decision tree ensemble prediction models are employed for prediction of PM10 in Thiruvanathapuram, the capital city of Kerala. Feature importance analysis of PM10 is also carried out in this study, since it plays a crucial role in understanding which variables significantly influence the model's prediction of PM10 values. Thiruvananthapuram demonstrates the traits of a city in development, with great potential for expansion and modernisation. Even though Thiruvananthapuram has comparatively lower pollution levels than other large Indian cities like Delhi, effective air quality management requires constant monitoring and preventative actions. So, to lower the particulate matter concentrations in Thiruvanathapuram and to lessen the health hazards related to air pollution in the city, ongoing research is needed to anticipate and monitor PM10 with greater accuracy. This research study is novel as it is the initial attempt in Kerala to predict PM10 levels using advanced machine learning techniques, specifically decision tree ensemble models. It incorporates feature importance analysis to identify key factors influencing PM10 concentrations, providing actionable insights for pollution control. By focusing on proactive air quality management in Thiruvananthapuram, a developing urban center with relatively low pollution levels, the research emphasizes accurate prediction and preventative strategies to support sustainable growth and health protection.

4. SYSTEM DESIGN

This study was conducted for the capital city of Kerala, Thiruvananthapuram. The ambient air quality monitoring station in the city is located at Plammoodu (Latitude: 8.51°N, Longitude: 76.94°E). Kerala State Pollution Control Board owns and operates the monitoring station.

Thiruvananthapuram generally exhibited better air quality in comparison to bigger Indian metropolitan areas such as Delhi and Mumbai. The city frequently ranks in the "Good" to "Moderate" categories on the air quality index (AQI) scale, with limited occurrences of severe pollution. The area's ambient air quality is negatively impacted by population density, personal vehicle usage, traffic congestion, housing construction, business and industrial units, and other factors. The data for the analysis was obtained from the Central Pollution Control Board's website. The data was collected for 914 days, from July 1, 2017 to December 31, 2019.

The dataset included daily values for seventeen features, encompassing date, air pollutant variables, and meteorological parameters, in addition to the target variable of daily PM10 values. The dataset included detailed temporal information with date values specifying the day, month, and year. Additionally, it contained daily concentration levels of various air pollutants, including Particulate Matter 2.5 (PM2.5), Nitric Oxide (NO), Nitrogen Dioxide (NO2), Nitrogen Oxides (NOx), Ammonia (NH3), Carbon Monoxide (CO), ozone, and Sulfur Dioxide (SO2). Wind Speed (WS), Wind Direction (WD), Ambient Temperature (AT), Relative Humidity (RH), Rainfall volume (RF), Solar Radiance (SR), and Buoyancy Pressure (BP) and Rack Temperature (Temp) were among the meteorological parameters included in the data set. The prediction models were developed in Python using the Colab Notebook of the Google Cloud Computing service environment. Figure 1 shows the plot of PM10 values over the study period from 2017 to 2019.

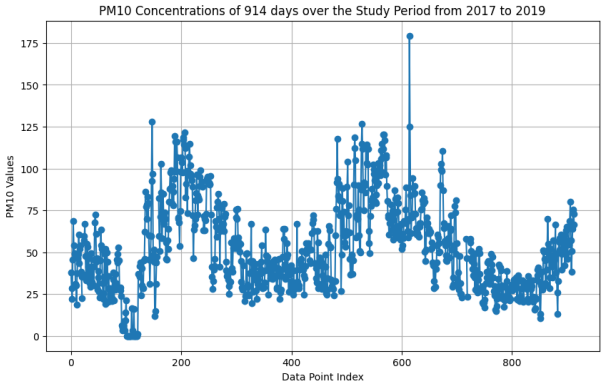

Figure 1. PM10 values over the years 2017 to 2019

Due to instrument malfunction, power loss, or communication problems, the dataset contained missing values. KNN imputation was used to handle missing values, which accounted for less than 5% of the dataset. KNN imputation was chosen because it effectively handles missing values by leveraging relationships across the sixteen air pollutant and meteorological variables while maintaining the dataset's multivariate structure (22,23). It captures the spatial and temporal dependencies contained in the dataset, resulting in accurate estimates influenced by weather and pollutant interactions. Furthermore, KNN is computationally efficient, adaptive, and produces more trustworthy imputations than simpler approaches such as

mean or median imputation (24,25). Rather than the simplistic approach of filling all the values with mean or median, it is a more powerful and useful strategy in which the missing value is estimated based on the mean of the neighbors (26). A distance from the missing values is defined in this method, which is known as the K parameter (27). The scikit-learn class KNNimputer was used here to fill in the missing values in a dataset and the parameter K was set to 15. KNN imputation with k=15 was chosen because it produced imputed values that preserve the original dataset's statistical properties, including mean, variance, and distribution shape (28). So, this empirical evaluation confirmed that the natural patterns in PM10, other air pollutants and atmospheric variables remain intact. Another reason for the choice of k=15 in KNN imputation was due to the balance of the bias-variance tradeoff. Smaller k values (e.g., k=1–5) cause high variance by overfitting to noise or outliers, while larger k values (e.g., k=30 or above) lead to high bias, over smoothing the data and losing local patterns (29,30). With 5% missing data in this dataset, k=15 ensured there were sufficient comparable neighbors to accurately impute values, even for features with multiple missing entries (31,32). The imputed dataset included 914 daily records of PM10 along with associated feature variables.

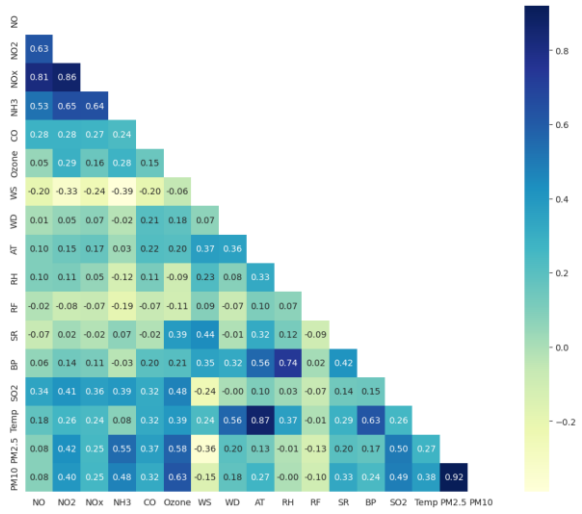

Figure 2. Correlation heatmap of PM10

A visualized heat map of the Pearson correlation coefficient between the input features and PM10 is shown in Figure 2, where blue represents positive correlation, while yellow indicates negative correlation. Those parameters that were highly positively correlated to PM10 are PM2.5, NH3, NO2, Ozone and SO2 (correlation coefficient > +0.40). The heatmap indicated that wind speed and rainfall were negatively correlated to PM10.

In this study, seven decision tree ensemble machine learning algorithms, namely, Random Forest, Extra Trees, Gradient Boosting, AdaBoost, LightGBM, XGBoost, and Histogram-Based Gradient Boosting were used for PM10 prediction. A random forest (RF) is a tree-based ensemble approach that uses a large number of weak decision tree learners that are

developed in parallel to reduce the model's bias and variance (33). RF is an ensemble technique that builds numerous decision trees during the training phase and subsequently averages their predictions to yield a more precise and stable outcome than a single tree (34). This method alleviates the overfitting issue in individual decision trees by averaging outcomes from several models. RF employs the bootstrap aggregating, wherein each tree is trained on a random subset of the data (35). This element of randomization is essential for ensuring that the trees remain uncorrelated, hence enhancing model performance.

The extremely randomized trees algorithm is a machine learning technique that was created as an extension of the random forest algorithm (36). It is also known as extra trees (ET). ET is an ensemble machine-learning algorithm that incorporates predictions from several decision trees to increase accuracy and reduce computational complexity (37). The ET method includes creating a randomized ensemble of trees and aggregating their predictions in a suitable manner, such as averaging in regression problems or majority voting in classification problems (38). ET uses the same concept as RF, training each base estimator with a random subset of features, but using the entire training dataset. However, when splitting the node, it chooses the best function and the corresponding value at random (39).

Gradient Boosting is an ensemble approach employed for predicting continuous outcomes by integrating several weak decision tree prediction models (40). This model is constructed incrementally, beginning with a basic model and progressively incorporating weak learners to enhance predictive accuracy (41,42). Each new model is trained to rectify the faults of its predecessors, concentrating on the residuals. This approach employs gradient descent to reduce the loss function (43).

Light Gradient Boosting Machine (LightGBM) is a sophisticated gradient boosting system, engineered for efficiency, precision, and scalability (44). It is an open-source framework created by Microsoft. It utilizes a leaf-wise growth technique, in contrast to traditional gradient boosting, which employs a level-wise growth approach (45). The leaf-wise growth strategy optimizes tree growth by prioritizing splits in the leaf with the greatest potential to minimize the loss function, enhancing precision and efficacy (46). LightGBM facilitates parallelization through two methods: feature-parallel training, wherein distinct computers handle various feature sets, and data-parallel training, which involves partitioning the dataset across many machines (47).

AdaBoost (Adaptive Boosting) is an ensemble learning method that integrates several weak learners to formulate a robust predictive model (48). This approach improves the efficacy of basic tree models by concentrating on the errors from prior iterations, rendering it especially helpful for regression tasks. It was created by Yoav Freund and Robert Schapire in 1996 (49). AdaBoost emphasizes weak learners with significant prediction mistakes, adaptively modifying their relevance during the training process (50).

XGBoost (eXtreme Gradient Boosting) is an innovative and widely utilized gradient boosting method, created by Tianqi Chen and Carlos Guestrin at the University of Washington in 2016 (51). XGBoost optimizes the boosting process by gradient boosting, L1 and L2 regularization, and parallel processing during tree construction (52). XGBoost is generally regarded for its flexibility and scalability, as it is engineered to accommodate enormous datasets and efficiently process billions of instances (53).

HistGradientBoosting (Histogram-based Gradient Boosting) is a gradient boosting framework that utilizes histogram-based methods to enhance the training of decision trees, rendering it especially appropriate for extensive datasets (54). The fundamental innovation of histogram-based gradient boosting resides in its data preparation and partitioning methodology (55). Histogram-based methods convert continuous features into discrete bins prior to the commencement of training (56).

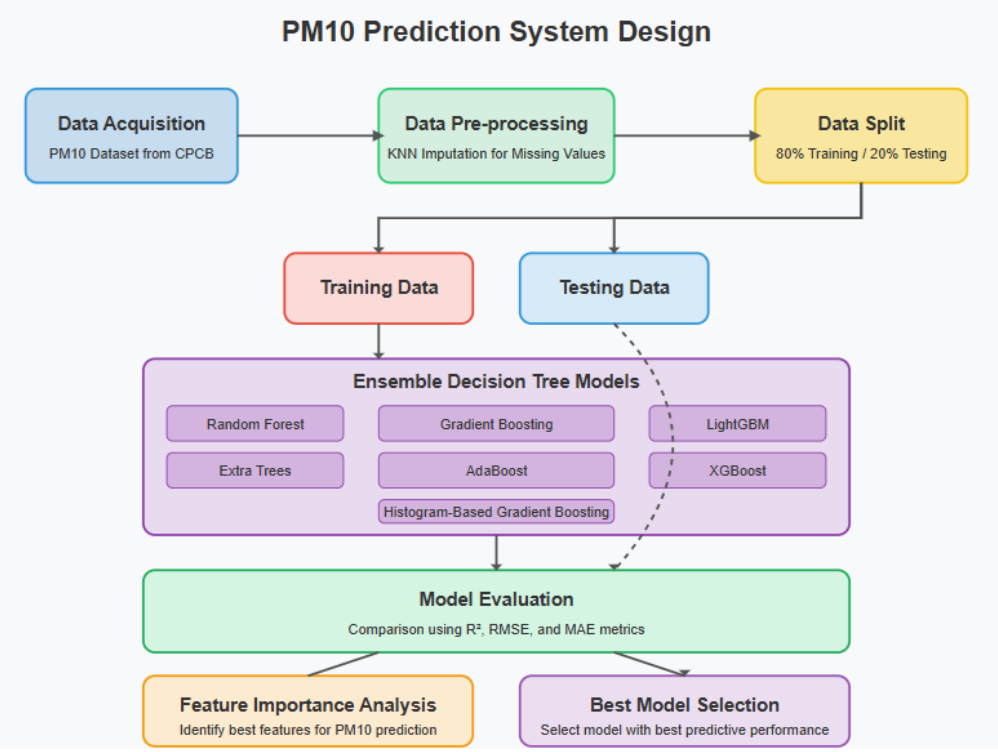

Figure 3. System design diagram

Feature importance analysis is an important process in machine learning that evaluates the contribution of each feature to the model's predictions by assigning scores based on their impact (57). This aids in determining which features are most influential and which are irrelevant, resulting in dimensionality reduction and improved model efficiency. It enhances interpretability by providing insights into the relationships between the input features and the target variable, making it easier to explain model decisions to stakeholders (58,59). Feature importance analysis in decision tree ensemble models entails determining how each feature

contributes to the model's predictions. These models determine feature relevance by calculating how much a feature decreases impurity (e.g., Gini impurity or entropy) or improves the loss function across all trees during splits (60). In scikit-learn Python library built-in characteristic named feature_importances provide access to the importance scores.

In this work, all the models were trained on 80% of the dataset and then tested on the remaining 20% to determine their ability to predict PM10 concentrations. The predicted PM10 values were compared to the actual values using three evaluation metrics: coefficient of determination (R^2), root mean square error (RMSE) and mean absolute error (MAE). The system design of the reseacrch work is diagrammatically represented in Figure 3.

5. EXPERIMENT AND ANALYSIS

The monthly variation trend of PM10 levels in Thiruvananthapuram is a significant element of air quality assessment, indicating the influence of many environmental conditions and anthropogenic activities on air pollution. Figure 4 showed that PM10 levels are elevated to the highest values during the wintertime, i.e., from November to February and after that a decline is reported during the summer season (from March to May). It should also be noted that PM10 concentrations were lower during the showery season of Southwest Monsoon which typically begins in early June and lasts until August. During the Northeast Monsoon season, which lasts from October to November, PM10 values decreased to their lowest levels. This was consistent with the results from the correlation heatmap, which indicated that rainfall and wind were negatively correlated with PM10 values. The heatmap in Figure 2 also demonstrated that temperature, wind direction, and solar radiance had a positive association with PM10 levels, indicating that increases in these variables were linked to elevated PM10 concentrations. Since the summer season in Kerala is characterized by high temperatures and high solar radiance, this correlation is in agreement with the summer monthly variation trend of PM10. From these observations, it can be concluded that the PM10 is truly season dependent.

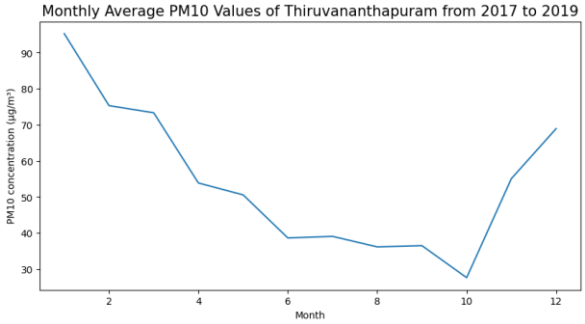

Figure 4. Monthly variation trend of PM10

Table 1 presents the comparison results of the daily predicted PM10 and actual PM10 values of the validation dataset for each model. When considering the prediction accuracy of PM10 for the validation data set, the

Extra Trees, Histogram Gradient Boosting and RF algorithms exhibited relatively higher accuracy results. Extra Trees demonstrated the superior prediction performance with the highest R^2 = 0.9397 and the lowest RMSE = 6.664 µg/m^3 and MAE = 4.950 µg/m^3 values. This indicates that approximately 93.97% of the variance in PM10 levels is explained by the Extra Trees model and also low error metrics suggests that the predictions made by Extra Trees are very close to the actual values, indicating high accuracy. The Histogram Gradient Boosting model demonstrated performance comparable to that of Extra Trees, achieving an R^2 of 0.9391, with the lower RMSE of 6.699 µg/m^3 and MAE of 5.008 µg/m^3. Following closely, the Random Forest model also exhibited slightly lower performance, with an R^2 of 0.9318, RMSE of 7.085 µg/m^3, and MAE of 5.186 µg/m^3. Gradient Booting model indicated comparable performance to the above-mentioned models (R^2 = 0.9251, RMSE = 7.430 µg/m^3 and MAE = 5.335 µg/m^3). In this study, all other boosting-based ensemble models (LightGBM, AdaBoost, and XGBoost) demonstrated lower performance. The XGBoost model recorded the least predictive performance, with the lowest R^2 value of 0.6539, along with the highest RMSE of 15.971 µg/m^3 and MAE of 13.101 µg/m^3.

Table 1. PM10 prediction accuracy

Model	R^2	RMSE (µg/m^3)	MAE (µg/m^3)
Random Forest	0.9318	7.085	5.186
Extra Trees	0.9397	6.664	4.950
Gradient Boosting	0.9251	7.430	5.335
LightGBM	0.7952	12.283	10.255
AdaBoost	0.8529	10.411	8.616
XGBoost	0.6539	15.971	13.101
HistGradientBoosting	0.9391	6.699	5.008

The scatter plots of the predicted PM10 values by the seven ensemble models using the validation data set, and the actual PM10 values are shown in Figure 5 to Figure 11 respectively. The red dotted line represents the 1:1 line and is used to assess how closely the predicted PM10 values match the actual values. For Extra Trees, Random Forest, HistogramGradient Boosting and Gradient Boosting models, the PM10 values strongly agree on the 1:1 line, whereas in the case of AdaBoost, LightGBM and XGBoost, the scatter is huge and so the accuracy of PM10 prediction of these models is low.

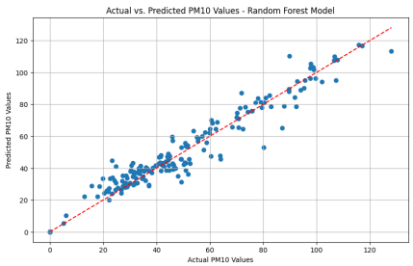

Figure 5. Scatter plot of predicted PM10 values by Random Forest model

After prediction performance analysis, the best four tree models selected were Extra Trees, HistogramGradientBoosting, RF and Gradient Boosting models. Feature importance analysis was conducted to determine the contribution of input variables to model predictions. For Extra Trees, Random Forest, and Gradient Boosting models, feature importance was calculated using their respective tree-based libraries in Python, leveraging feature_importances property. For the Histogram Gradient Boosting model, permutation feature importance was applied. This feature importance analysis approach provides insights into key factors influencing PM10 levels and aids in refining predictive accuracy.

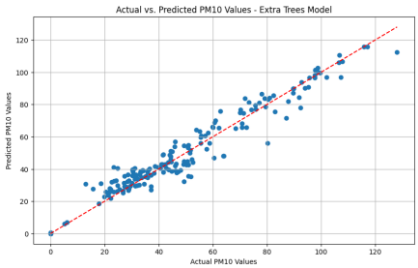

Figure 6. Scatter plot of predicted PM10 values by Extra Trees model

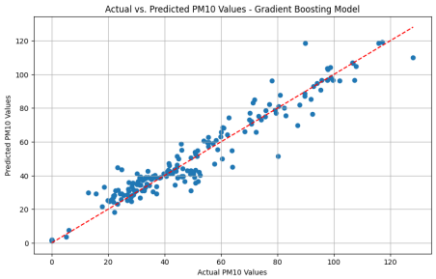

Figure 7. Scatter plot of predicted PM10 values by GradientBoosting model

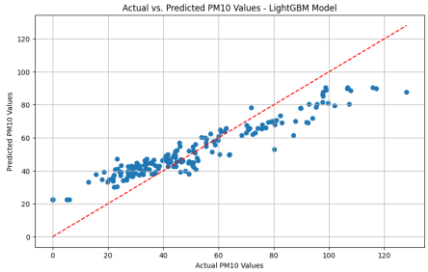

Figure 8. Scatter plot of predicted PM10 values by LightGBM model

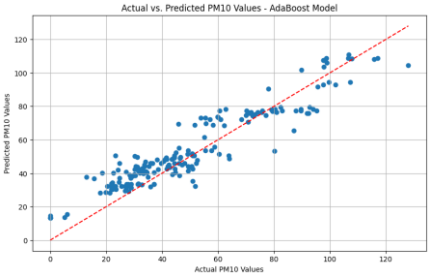

Figure 9. Scatter plot of predicted PM10 values by AdaBoost model

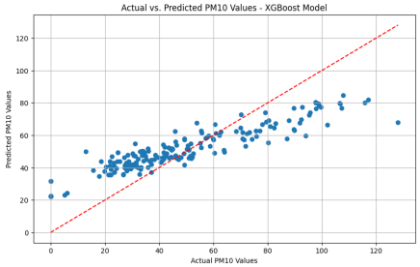

Figure 10. Scatter plot of predicted PM10 values by XGBoost model

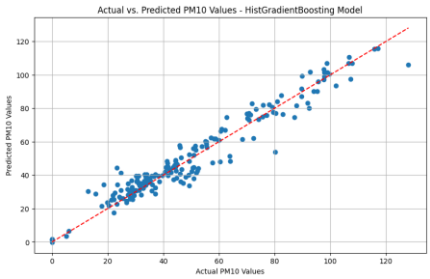

Figure 11. Scatter plot of predicted PM10 by Histogram GradientBoosting model

Table 2. Feature Importance analysis in PM10 prediction

Extra Trees	Histogram GradientBoosting	RF	Gradient Boosting
SO2	SO2	SO2	SO2
NH3	CO	WS	CO
BP	NO2	CO	WS
WS	WS	BP	BP
NO2	SR	NO	NO
CO	NH3	NO2	NH3
WD	NOx	NOx	WD
SR	RF	WD	RF
Ozone	BP	NH3	NO2
RH	PM2.5	RF	Ozone

The most important ten features affecting the PM10 prediction using these models are shown in Table 2. The feature analysis process highlights that SO2 is the most significant feature consistently identified by all four models as influencing PM10 predictions, underscoring its crucial role in air pollution dynamics. Other important air pollutants include CO, NO2, and NH3, which contribute to particulate matter formation through atmospheric reactions. Additionally, meteorological factors such as wind speed, rainfall, and barometric pressure significantly impact PM10 levels by affecting pollutant dispersion and accumulation.

6. CONCLUSION

Forecasting PM10 concentrations is vital for environmental monitoring and public health management systems. Since air pollutants and meteorological conditions are linked in a complex way, air quality modelling is a difficult process. Hence, machine learning techniques can be applied to

improve the modeling of air pollutant concentrations. In this study, the feasibility of seven decision tree ensemble models in predicting the particulate matter PM10 concentrations of Thiruvananthapuram city is investigated. Extra Trees model exhibited superior prediction model with highest R2 value and lowest RMSE and MAE error metric values. The next best performing prediction models are HistogramGradientBoosting, Random Forest and Gradient Boosting models. After prediction process, the feature importance analysis is done to determine how much each input feature contributes to the predictions made by the above four best models. The most important feature influencing the PM10 prediction is SO2. The other air contaminants that affect PM10 prediction are CO, NO2 and NH3, while the prominent meteorological features affecting PM10 are wind speed, rainfall and barometric pressure. This study is a pioneering effort in Thiruvananthapuram, employing decision tree ensemble models for PM10 prediction. It improves prediction accuracy while identifying key pollution drivers through feature importance analysis, offering actionable insights for targeted mitigation strategies.

Acknowledgments

The authors would like to thank the Central Pollution Control Board for providing the dataset of this research work.

REFERENCES

[1] Wu X, Wang Y, He S, Wu Z. **PM 2.5/PM 10 ratio prediction based on a long short-term memory neural network in Wuhan, China**. *Geoscientific Model Development*. 2020;13(3):1499–511.

[2] Tong X, Ho JMW, Li Z, Lui KH, Kwok TC, Tsoi KK, et al. **Prediction model for air particulate matter levels in the households of elderly individuals in Hong Kong**. *Science of The Total Environment*. 2020;717:135323.

[3] Brunekreef B, Holgate ST. **Air pollution and health**. *The Lancet*. 2002 Oct 19;360(9341):1233–42.

[4] Pope III CA. **Lung Cancer, Cardiopulmonary Mortality, and Long-term Exposure to Fine Particulate Air Pollution**. *JAMA*. 2002 Mar 6;287(9):1132.

[5] Samet JM, Zeger SL, Dominici F, Curriero F, Coursac I, Dockery DW, et al. **The National Morbidity, Mortality, and Air Pollution Study. Part II: Morbidity and mortality from air pollution in the United States**. *Res Rep Health Eff Inst*. 2000 Jun;94(Pt 2):5–70.

[6] Huffman MD, Prabhakaran D, Osmond C, Fall CHD, Tandon N, Lakshmy R, et al. **Incidence of Cardiovascular Risk Factors in an Indian Urban Cohort**. *J Am Coll Cardiol*. 2011 Apr 26;57(17):1765–74.

[7] Gakidou E, Afshin A, Abajobir AA, Abate KH, Abbafati C, Abbas KM, et al. **Global, regional, and national comparative risk assessment of 84**

behavioural, environmental and occupational, and metabolic risks or clusters of risks, 1990–2016. *The Lancet*. 2017 Sep 16;390(10100):1345–422.

[8] Analitis A, Barratt B, Green D, Beddows A, Samoli E, Schwartz J, et al. **Prediction of PM2.5 concentrations at the locations of monitoring sites measuring PM10 and NOx, using generalized additive models and machine learning methods**. *Atmospheric Environment*. 2020 Nov 1;240:117757.

[9] Xi X, Wei Z, Xiaoguang R, Yijie W, Xinxin B, Wenjun Y, et al. **A comprehensive evaluation of air pollution prediction improvement by a machine learning method**. In: *2015 IEEE International Conference on Service Operations And Logistics, And Informatics (SOLI)*. IEEE; 2015. p. 176–81.

[10] Hooyberghs J, Mensink C, Dumont G, Fierens F, Brasseur O. **A neural network forecast for daily average PM10 concentrations in Belgium**. *Atmospheric Environment*. 2005 Jun 1;39(18):3279–89.

[11] Perez P, Reyes J. **Prediction of maximum of 24-h average of PM10 concentrations 30h in advance in Santiago, Chile**. *Atmospheric Environment*. 2002 Sep 1;36(28):4555–61.

[12] Brokamp C, Jandarov R, Hossain M, Ryan P. **Predicting Daily Urban Fine Particulate Matter Concentrations Using a Random Forest Model**. *Environmental Science & Technology*. 2018 Apr 3;52(7):4173–9.

[13] Weizhen H, Zhengqiang L, Yuhuan Z, Hua X, Ying Z, Kaitao L, et al. **Using support vector regression to predict PM10 and PM2.5**. *IOP Conference Series: Earth and Environmental Science*. 2014 Mar;17:012268.

[14] Wang P, Liu Y, Qin Z, Zhang G. **A novel hybrid forecasting model for PM10 and SO2 daily concentrations**. *Science of The Total Environment*. 2015 Feb 1;505:1202–12.

[15] Park J, Yoo S, Kim K, Gu Y, Lee K, Son U. **PM10 density forecast model using long short term memory**. In: *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE; 2017. p. 576–81.

[16] Suleiman A, Tight MR, Quinn AD. **Applying machine learning methods in managing urban concentrations of traffic-related particulate matter (PM10 and PM2.5)**. *Atmospheric Pollution Research*. 2019 Jan 1;10(1):134–44.

[17] Ibrir A, Kerchich Y, Hadidi N, Merabet H, Hentabli M. **Prediction of the concentrations of PM1, PM2.5, PM4, and PM10 by using the hybrid dragonfly-SVM algorithm**. *Air Quality, Atmosphere & Health*. 2021 Mar 1;14(3):313–23.

[18] Saini J, Dutta M, Marques G. **A novel application of fuzzy inference system optimized with particle swarm optimization and genetic algorithm for PM10 prediction**. *Soft Computing*. 2022 Sep 1;26(18):9573–86.

[19] Kim BY, Lim YK, Cha JW. **Short-term prediction of particulate matter (PM10 and PM2.5) in Seoul, South Korea using tree-based machine learning algorithms**. *Atmospheric Pollution Research*. 2022 Oct 1;13(10):101547.
[20] Guo Q, He Z, Wang Z. **Prediction of Hourly PM2.5 and PM10 Concentrations in Chongqing City in China Based on Artificial Neural Network**. *Aerosol and Air Quality Research*. 2023;23(6):220448.
[21] Nasabpour Molaei S, Salajegheh A, Khosravi H, Nasiri A, Ranjbar Saadat Abadi A. **Prediction of hourly PM10 concentration through a hybrid deep learning-based method**. *Earth Science Informatics*. 2024 Feb 1;17(1):37–49.
[22] Erhan L, Di Mauro M, Anjum A, Bagdasar O, Song W, Liotta A. **Embedded Data Imputation for Environmental Intelligent Sensing: A Case Study**. *Sensors*. 2021 Nov 23;21(23):7774.
[23] Saeipourdizaj P, Sarbakhsh P, Gholampour A. **Application of imputation methods for missing values of PM10 and O3 data**. *Environmental Health Engineering and Management Journal*. 2021 Aug 10;8(3):215–26.
[24] Oktaviani ID, Putrada AG. **KNN imputation to missing values of regression-based rain duration prediction on BMKG data**. *JURNAL INFOTEL*. 2022 Nov 1;14(4):249–54.
[25] Juna A, Umer M, Sadiq S, Karamti H, Eshmawi AA, Mohamed A, et al. **Water Quality Prediction Using KNN Imputer and Multilayer Perceptron**. *Water*. 2022 Jan;14(17):2592.
[26] Zhang S. **Nearest neighbor selection for iteratively kNN imputation**. *Journal of Systems and Software*. 2012 Nov 1;85(11):2541–52.
[27] Kim SL, D. **Imputation method for missing data based on KNN and pattern consistency index in microarray data**. *The Korean Data & Information Science Society*. 2018 Sep 30;(5):1179–87.
[28] Sundararajan A, Sarwat AI. **Evaluation of Missing Data Imputation Methods for an Enhanced Distributed PV Generation Prediction**. In: *Proceedings of the Future Technologies Conference (FTC) 2019*. *Springer International Publishing*; 2020. p. 590–609.
[29] Alianso AS, Syafaah L, Faruq A. **K-nearest neighbor imputation for missing value in hepatitis data**. *AIP Conference Proceedings*. 2022 Jul 25;2453(1):020057.
[30] Atik SO, Atik ME. **Optimal band selection using explainable artificial intelligence for machine learning-based hyperspectral image classification**. *Journal of Applied Remote Sensing*. 2024 Jul;18(4):042604.
[31] Beretta L, Santaniello A. **Nearest neighbor imputation algorithms: a critical evaluation**. *BMC Medical Informatics and Decision Making*. 2016 Jul 25;16(3):74.
[32] Murti DMP, Pujianto U, Wibawa AP, Akbar MI. **K-Nearest Neighbor (K-NN) based Missing Data Imputation**. In: *2019 5th International*

Conference on Science in Information Technology (ICSITech). IEEE; 2019. p. 83–8.
[33] Breiman L. **Random Forests**. *Machine Learning*. 2001 Oct 1;45(1):5–32.
[34] Babar B, Luppino LT, Boström T, Anfinsen SN. **Random forest regression for improved mapping of solar irradiance at high latitudes**. *Solar Energy*. 2020 Mar 1;198:81–92.
[35] Babu S, Thomas B. **A survey on air pollutant PM2.5 prediction using random forest model**. *Environmental Health Engineering and Management Journal*. 2023 Mar 10;10(2):157–63.
[36] Geurts P, Ernst D, Wehenkel L. **Extremely randomized trees**. *Machine Learning*. 2006 Apr 1;63(1):3–42.
[37] Yarveicy H, Ghiasi MM. **Modeling of gas hydrate phase equilibria: Extremely randomized trees and LSSVM approaches**. *Journal of Molecular Liquids*. 2017 Oct 1;243:533–41.
[38] Nistane V, Harsha S. **Performance evaluation of bearing degradation based on stationary wavelet decomposition and extra trees regression**. *World Journal of Engineering*. 2018 Jan 1;15(5):646–58.
[39] Basu V. **Prediction of Stellar Age with the Help of Extra-Trees Regressor in Machine Learning**. *Social Science Research Network*. 2020 Mar.
[40] Zhang Z, Zhao Y, Canes A, Steinberg D, Lyashevska O, ABDCTC Group. **Predictive analytics with gradient boosting in clinical medicine**. *Annals of Translational Medicine*. 2019 Apr;7(7):152.
[41] Friedman JH. **Greedy Function Approximation: A Gradient Boosting Machine**. *The Annals of Statistics*. 2001;29(5):1189–232.
[42] Li X, Li W, Xu Y. **Human Age Prediction Based on DNA Methylation Using a Gradient Boosting Regressor**. *Genes*. 2018 Sep;9(9):424.
[43] Kujawska J, Kulisz M, Oleszczuk P, Cel W. **Machine Learning Methods to Forecast the Concentration of PM10 in Lublin, Poland**. *Energies*. 2022 Jan;15(17):6428.
[44] Sun X, Liu M, Sima Z. **A novel cryptocurrency price trend forecasting model based on LightGBM**. *Finance Research Letters*. 2020 Jan 1;32:101084.
[45] Liu Y, Zhu R, Zhai S, Li N, Li C. **Lithofacies identification of shale formation based on mineral content regression using LightGBM algorithm**. *Energy Science & Engineering*. 2023;11(11):4256–72.
[46] Xuan L, Lin Z, Liang J, Huang X, Li Z, Zhang X, et al. **Prediction of resilience and cohesion of deep-fried tofu by ultrasonic detection and LightGBM regression**. *Food Control*. 2023 Dec 1;154:110009.
[47] Shehadeh A, Alshboul O, Al Mamlook RE, Hamedat O. **Machine learning models for predicting the residual value of heavy construction equipment**. *Automation in Construction*. 2021 Sep 1;129:103827.
[48] Schapire RE. **Explaining AdaBoost**. In: *Empirical Inference: Festschrift in Honor of Vladimir N Vapnik*. *Springer*; 2013. p. 37–52.

[49] Freund Y, Schapire RE. **A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting**. *Journal of Computer and System Sciences*. 1997 Aug 1;55(1):119–39.
[50] Koduri SB, Gunisetti L, Ramesh CR, Mutyalu KV, Ganesh D. **Prediction of crop production using adaboost regression method**. *Journal of Physics: Conference Series*. 2019 May;1228(1):012005.
[51] Chen T, Guestrin C. **XGBoost: A Scalable Tree Boosting System**. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. *Association for Computing Machinery*; 2016. p. 785–94.
[52] Abbasi RA, Javaid N, Ghuman MNJ, Khan ZA, Ur Rehman S, Amanullah. **Short Term Load Forecasting Using XGBoost**. In: *Web, Artificial Intelligence and Network Applications*. *Springer International Publishing*; 2019. p. 1120–31.
[53] Lartey B, Homaifar A, Girma A, Karimoddini A, Opoku D. **XGBoost: a tree-based approach for traffic volume prediction**. In: *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE; 2021. p. 1280–6.
[54] Guryanov A. **Histogram-Based Algorithm for Building Gradient Boosting Ensembles of Piecewise Linear Decision Trees**. In: *Analysis of Images, Social Networks and Texts*. *Springer International Publishing*; 2019. p. 39–50.
[55] Tamim Kashifi M, Ahmad I. **Efficient Histogram-Based Gradient Boosting Approach for Accident Severity Prediction With Multisource Data**. *Transportation Research Record*. 2022 Jun 1;2676(6):236–58.
[56] Hossain SMdM, Deb K. **Plant Leaf Disease Recognition Using Histogram Based Gradient Boosting Classifier**. In: *Intelligent Computing and Optimization*. *Springer International Publishing*; 2021. p. 530–45.
[57] Alhams A, Abdelhadi A, Badri Y, Sassi S, Renno J. **Enhanced Bearing Fault Diagnosis Through Trees Ensemble Method and Feature Importance Analysis**. *Journal of Vibration Engineering & Technology*. 2024 Dec 1;12(1):109–25.
[58] Feng DC, Wang WJ, Mangalathu S, Hu G, Wu T. **Implementing ensemble learning methods to predict the shear strength of RC deep beams with/without web reinforcements**. *Engineering Structures*. 2021 May 15;235:111979.
[59] Kanaparthi V. **Credit Risk Prediction using Ensemble Machine Learning Algorithms**. In: *2023 International Conference on Inventive Computation Technologies (ICICT)*. IEEE; 2023. p. 41–7.
[60] Sun Y, Li G, Zhang N, Chang Q, Xu J, Zhang J. **Development of ensemble learning models to evaluate the strength of coal-grout materials**. *International Journal of Mining Science and Technology*. 2021 Mar 1;31(2):153–62.