www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 09, September 2025, pp : 326-331

e-ISSN : 2583-1062
Impact Factor : 7.001

FORECASTING HOURLY $PM_{2.5}$ CONCENTRATIONS USING STL DECOMPOSITION WITH MACHINE LEARNIG AND DEEP LEARNING ENSEMBLE MODEL

Awanindra Kumar Singh[1]

[1]M.Tech. Scholar, Department Of Civil Engineering, Institute Of Engineering & Technology, Lucknow, Uttar Pradesh, India.

DOI: https://www.doi.org/10.58257/IJPREMS43766

ABSTRACT

Fine particulate matter ($PM_{2.5}$) poses significant health risks due to its ability to penetrate the respiratory system and bloodstream, originating from anthropogenic and natural sources. Accurate hourly forecasting is essential for public health warnings and emission control. This study develops a hybrid model for hourly $PM_{2.5}$ concentration prediction using Seasonal-Trend decomposition via LOESS (STL) to separate data into trend, seasonal, and residual components. Data from Talkatora, Lucknow (India), collected via the Central Pollution Control Board, underwent preprocessing including missing value imputation and outlier removal. The trend component was forecasted with Linear Regression (LR) using 24-hour lags, the seasonal component with eXtreme Gradient Boosting (XGB) also incorporating 24-hour lags, and the residual with Long Short-Term Memory (LSTM) neural network (64 cells, Adam optimizer, MSE loss). Forecasts were aggregated for final predictions. The model was compared against standalone LR, XGB, LSTM, and STL variants using MAE, RMSE, Pearson's correlation coefficient r, and R^2 on test data. Results showed the hybrid STL-XGB-LR-LSTM model outperformed others, achieving MAE of 8.4736, RMSE of 13.0953, r of 0.9541, and R^2 of 0.9098, indicating superior accuracy in capturing temporal patterns. This approach enhances $PM_{2.5}$ forecasting for proactive environmental management.

Keywords: Forecasting; $PM_{2.5}$, STL Decomposition, Machine Learning, Deep Learning.

1. INTRODUCTION

Fine particulate matter ($PM_{2.5}$), comprising particles with an aerodynamic diameter less than 2.5 micrometres, is a critical air pollutant with significant implications for public health and environmental sustainability. Due to their small size, $PM_{2.5}$ particles penetrate deep into the respiratory system and bloodstream, exacerbating global morbidity and mortality. Accurate forecasting of $PM_{2.5}$ concentrations is vital for mitigating exposure risks, issuing timely public health warnings, and shaping effective policy interventions to reduce emissions and protect vulnerable populations [1]. $PM_{2.5}$ originates from both primary and secondary sources. Primary emissions are generated by anthropogenic activities, including vehicular exhaust, industrial processes, coal combustion, and biomass burning, as well as natural sources such as dust storms and wildfires. Secondary $PM_{2.5}$ forms through atmospheric chemical reactions involving precursor gases like sulfur dioxide (SO_2), nitrogen oxides (NOx), and volatile organic compounds (VOCs). In regions like central Bangladesh, studies using positive matrix factorization (PMF) and self-organizing maps (SOM) have identified significant correlations between $PM_{2.5}$ and pollutants such as NO2, black carbon, and methane, highlighting contributions from brick kilns, urban density, and traffic [2]. These sources are particularly pronounced during dry seasons, intensifying regional pollution levels. The health effects of $PM_{2.5}$ are extensive, driven by mechanisms including oxidative stress, inflammation, cytokine release, DNA damage, altered gene expression, and apoptosis. Short-term exposure is associated with aggravated respiratory symptoms, cardiovascular events, and increased hospital admissions, while long-term exposure contributes to chronic conditions such as cardiopulmonary diseases, neurological disorders, and elevated mortality risks [3]. Time-series studies indicate that $PM_{2.5}$ has a stronger association with respiratory morbidity, such as asthma exacerbations and pneumonia, than cardiovascular outcomes in short-term scenarios, with children, the elderly, and individuals with pre-existing conditions being particularly vulnerable [4]. Due to the aforementioned health effects of $PM_{2.5}$ there is an urgent need for effective mitigation strategies. Forecasting $PM_{2.5}$ concentrations enables proactive measures to prevent exceedances of safety thresholds, such as the World Health Organization’s 15 µg/m³ daily limit.

Various studies have demonstrated the effectiveness of machine learning models (MLMs) in forecasting $PM_{2.5}$ concentrations. For example, MLMs have been used to predict $PM_{2.5}$ levels in Quito, Ecuador [5]. In Taiwan, one study found that Gradient Boosting Regressor (GBR) outperformed several other models, including Linear Regression, Random Forest (RF), and K-Nearest Neighbours (KNN) [6]. Deep learning models have also been extensively used for this purpose. A Multi-Layered Perceptron (MLP) has been used for both PM_{10} and $PM_{2.5}$

predictions [7]. In South Korea, an interpolated Convolutional Neural Network (CNN) was employed for hourly forecasts, with interpolation being used to manage imbalanced spatial data [8].

Another approach to improving forecast accuracy is using time series decomposition techniques, which break down data into components. Ensemble Empirical Mode Decomposition (EEMD) has been used to decompose $PM_{2.5}$ data into multiple Intrinsic Mode Functions (IMFs), which were then forecasted with a General Regression Neural Network (GRNN) [9]. A combination of EEMD with Phase Space Reconstruction (PSR) and Least Square Support Vector Machine (LSSVM) has also been used for enhanced prediction [10].

Recent studies have improved $PM_{2.5}$ forecasting by integrating STL decomposition with machine learning. The HISTCP framework, for instance, uses STL to break down $PM_{2.5}$ data into trend, seasonal, and residual components, processed via moving average smoothing, least squares fitting, and an optimized linear dendritic model, respectively. Tested on data from five Chinese cities (2010–2015), HISTCP outperformed models like ANFIS and Transformers in MSE and R^2 [11]. Another study combined STL decomposition with CNN, BiLSTM, and attention mechanisms, optimized via Bayesian methods, achieving up to 30% lower RMSE on Delhi data (2019–2023) compared to non-decomposed models [12].

Both above STL decomposition models were developed for daily average $PM_{2.5}$ concentrations which does not consider the variations in the $PM_{2.5}$ concentrations throughout the day. This paper aims to address this issue by developing a forecasting model for hourly $PM_{2.5}$ concentrations based on STL decomposition and machine learning models.

2. METHODOLOGY

First hourly $PM_{2.5}$ data was collected from the Central Pollution Control Board (CPCB) for the location Talkatora, Lucknow, Uttar Pradesh (India). After data collection, data preprocessing was done. Then STL decomposition was applied on the preprocessed hourly $PM_{2.5}$ data to decompose the data into seasonal, trend and residual components. The seasonal component was then forecasted using eXtreme Gradient Boosting (XGB) with lag feature model, trend component was forecasted using Linear Regression (LR) with lag feature model and the residual component was forecasted using Long Short-Term Memory (LSTM) neural network. The individual forecasted components were then aggregated to get the final forecast values. The models were built and trained on the Google's Colab platfrom. The proposed model was evaluated against STL decomposition-XGB, STL decomposition-LR, STL decomposition-LSTM, XGB, LR and LSTM models in terms of mean absolute error (MAE), root mean squared error (RMSE), Pearson's correlation coefficient (r) and coefficient of determination (R^2).

3. MODELING AND ANALYSIS

Data Collection and Preprocessing

Hourly $PM_{2.5}$ data was collected from Central Pollution Control Board (CPCB) for the location Talkatora, Lucknow, Uttar Pradesh (India). Data preprocessing such as filling missing values with mean, removing and replacing outliers using linear interpolation was performed. Then on the processed data STL decomposition was applied to decompose the data into seasonal, trend and residual components.

STL Decomposition

Seasonal-Trend decomposition using Locally Estimated Scatterplot Smoothing (LOESS), commonly known as STL decomposition, is a robust statistical method for analysing time series data by separating it into three distinct components: trend, seasonal, and residual. This approach is particularly valuable in environmental sciences, such as $PM_{2.5}$ forecasting, where time series exhibit complex patterns driven by seasonal variations, long-term trends, and irregular fluctuations. STL decomposition enables enhanced modelling by isolating these components for tailored analysis, improving the accuracy of predictive models [11].

The trend component captures the long-term progression or direction in the data, such as a gradual increase in $PM_{2.5}$ concentrations due to urbanization. The seasonal component reflects recurring patterns, like seasonal pollution peaks during winter months due to meteorological factors or biomass burning. The residual component encompasses irregular, short-term fluctuations not explained by trend or seasonality, such as sudden spikes from wildfires. STL employs LOESS, a non-parametric regression technique, to smooth the time series iteratively, ensuring flexibility in handling non-linear patterns and robustness against outliers [13].

The STL process begins by detrending the data to isolate seasonality, followed by smoothing to estimate the seasonal component. The trend is then derived by removing the seasonal component and smoothing the remainder. Finally, residuals are calculated as the difference between the original series and the sum of trend and seasonal components.

This additive decomposition assumes the time series can be expressed as: Y(t) = T(t) + S(t) + R(t), where Y(t) is the observed data, T(t) is the trend, S(t) is the seasonal component, and R(t) is the residual [1].

eXtreme Gradient Boosting (XGB)

XGB is a powerful machine learning algorithm that leverages gradient boosting to build predictive models, particularly effective for time series forecasting tasks like $PM_{2.5}$ prediction. It constructs an ensemble of decision trees iteratively, where each tree corrects the errors of its predecessors by minimizing a loss function through gradient descent. XGB's strengths include its scalability, handling of non-linear relationships, and robustness to noisy data, making it suitable for complex environmental datasets [14]. All the XGB models developed for this study uses 24-hour lag features.

In time series forecasting, lag features—past observations used as predictors—are critical for capturing temporal dependencies. For instance, in $PM_{2.5}$ forecasting, lag features might include $PM_{2.5}$ concentrations from previous hours or days, reflecting patterns like diurnal cycles or pollution persistence. These lagged values are incorporated as input variables in XGB, enabling the model to learn how historical data influences future outcomes. The algorithm optimizes feature importance, assigning higher weights to lags with stronger predictive power, such as recent $PM_{2.5}$ levels.

Linear Regression (LR)

Linear Regression (LR) is a statistical method that models the relationship between a dependent variable and one or more independent variables using a linear equation: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n + \varepsilon$, where β coefficients represent the impact of predictors X on outcome Y, and ε is the error term. In time series forecasting, such as $PM_{2.5}$ concentration prediction, LR assumes linearity and independence of observations, but temporal data often violates this due to autocorrelation [13].

To address this, lag features—past values of the dependent or independent variables—are incorporated as predictors. This transforms LR into an autoregressive model, capturing temporal dependencies: $Y_t = \beta_0 + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + ... + \beta_k X_{t-l} + \varepsilon_t$, where t denotes time, and lags (e.g., t_1, t_2) reflect historical influences. In forecasting problems, lag features enhance accuracy by accounting for seasonal and trend patterns. All the LR models used for this study were built with lag feature of 24-hour lag.

LSTM Neural Network

Long Short-Term Memory (LSTM) networks are an advanced form of Recurrent Neural Networks (RNNs) designed to model sequential data with long-term dependencies, overcoming key limitations of standard RNNs. Traditional RNNs process sequences by updating a hidden state over time, suitable for tasks like time-series analysis or natural language processing. However, they struggle with vanishing and exploding gradients during backpropagation, which disrupts learning over extended sequences [15]. These gradient issues cause information from earlier time steps to either fade or amplify uncontrollably, limiting RNNs' ability to capture distant dependencies [16]. LSTMs address these challenges through a sophisticated architecture featuring a cell state, which acts as a persistent memory channel to retain information across long sequences [17]. The cell state is regulated by three specialized gates—forget, input, and output—that control the flow and retention of information using sigmoid and tanh activation functions.

The forget gate decides which parts of the cell state to discard, calculated as $f_t = \sigma\ (W_f \cdot [h_t - 1, x_t] + b_f)$, where σ denotes the sigmoid function, h_{t-1} is the prior hidden state, x_t is the current input, and W_f, b_f represent weights and biases [15]. The input gate determines what new information to incorporate, using a sigmoid layer $i_t = \sigma\ (W_i \cdot [h_{t-1}, x_t] + b_i)$ and a *tanh* layer for candidate values $\check{C}_t = tanh\ (W_C \cdot [h_{t-1}, x_t] + b_C)$. The cell state updates via $C_t = f_t \odot C_{t-1} + i_t \odot C_t$, with $\odot$ indicating element-wise multiplication. The output gate filters the cell state to produce the hidden state: $o_t = \sigma\ (W_o \cdot [h_{t-1}, x_t] + b_o)$, and $h_t = o_t \odot tanh\ (C_t)$ [16]. This gated structure enables LSTMs to selectively retain or discard information, making them effective for tasks requiring long-range context.

All the LSTM models used for this study were built with the same hyper parameters, i.e. number of LSTM layers is one, number of LSTM cells in each layer is 64, adam optimiser, mean squared error as loss function and early stopping with patience level 10.

Performance Evaluation Metrics

By comparing the performance errors, the models' prediction performance was assessed. Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Coefficient of Determination (R^2) were the quantitative evaluation metrics that were employed. Additionally, the linear relationship between the predicted and actual values was measured using Pearson's Coefficient of Correlation (r). The following are the mathematical formulas for RMSE, MAE, R^2 and r:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|Y_i - X_i| \tag{1}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y_i - X_i)^2} \tag{2}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(Y_i - X_i)^2}{\sum_{i=1}^{N}((\frac{1}{N}\sum_{i=1}^{N}X_i) - X_i)^2} \tag{3}$$

$$r = \frac{\sum_{i=1}^{N}(Y_i - \frac{1}{N}\sum_{i=1}^{N}Y_i)(X_i - \frac{1}{N}\sum_{i=1}^{N}X_i)}{\sqrt{\sum_{i=1}^{N}(Y_i - \frac{1}{N}\sum_{i=1}^{N}Y_i)^2}\sqrt{\sum_{i=1}^{N}(X_i - \frac{1}{N}\sum_{i=1}^{N}X_i)^2}} \tag{4}$$

Where Y is the predicted value and X is true value and N is the number of observations.

4. RESULTS AND DISCUSSION

To determine the kind of seasonality present in the $PM_{2.5}$ dataset time series graph of the dataset was plotted which can be seen in Figure 1 below. From the graph it can be clearly inferred that yearly seasonality is present in the $PM_{2.5}$ dataset of the location Talkatora.

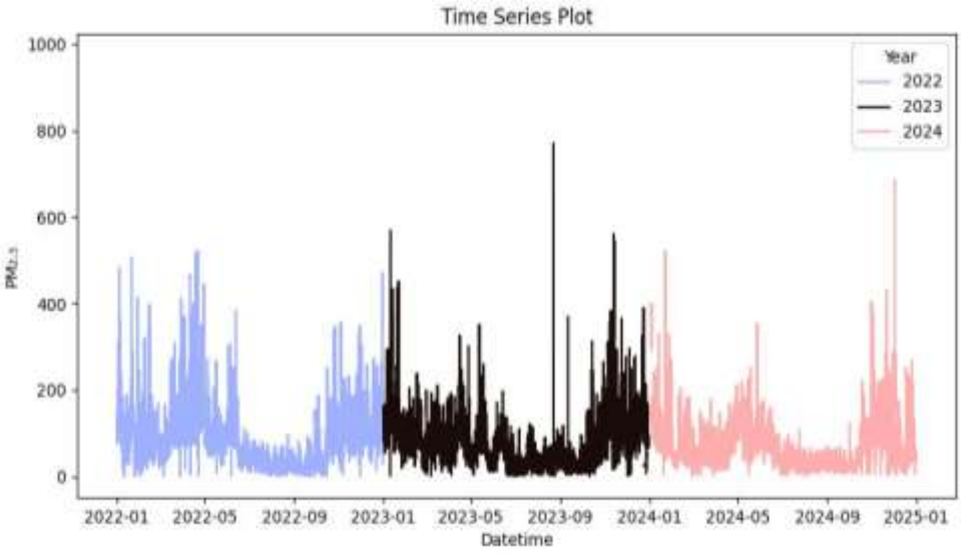

Figure 1: $PM_{2.5}$ Time Series Plot of Talkatora

After determining the seasonality, STL decomposition was used to decompose the $PM_{2.5}$ data into seasonal, trend and residual components. The graph of STL decomposition components along with the observed values is shown in Figure 2.

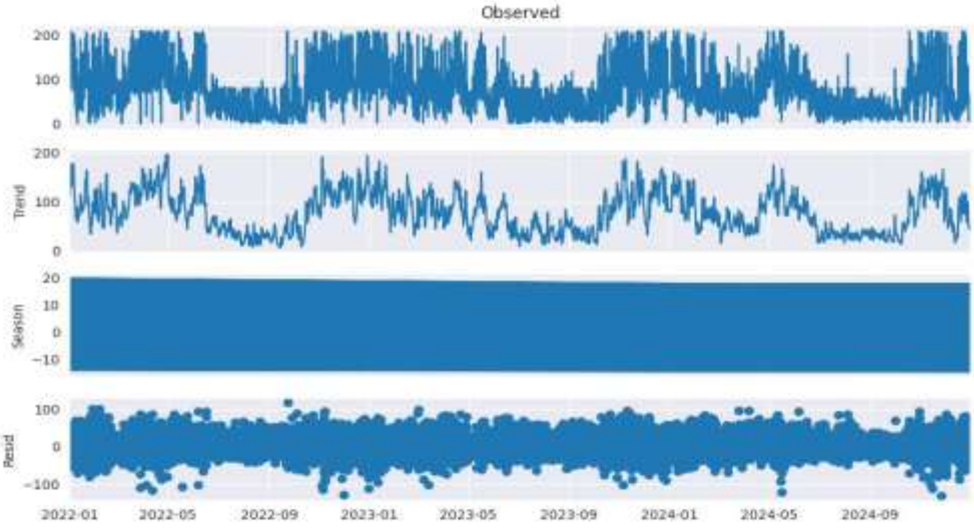

Figure 2: STL Decomposition Plot

The performance metrices of all the models is shown in table 1 below.

Table 1: Performance Evaluation Metrics for all the Models

Performance Metrics	STL-XGB-LR-LSTM	STL-LSTM	STL-XGB	STL-LR	LSTM	XGB	LR
MAE	8.4736	8.6330	10.1189	10.3234	9.0028	10.6871	10.4723
RMSE	13.0953	13.3437	15.2220	15.4950	14.0719	16.4026	16.4075
r	0.9541	0.9535	0.9492	0.9473	0.9489	0.9407	0.9407
R^2	0.9098	0.9090	0.9009	0.8973	0.8988	0.8849	0.8849

From the above table several inferences can be made. First all the models showed good performance in forecasting $PM_{2.5}$. The standalone LSTM model provided a good baseline for forecasting, showing a relatively strong correlation between true and predicted values. The LR model, using a simple lagged feature, also demonstrated a strong positive linear relationship and a reasonable R^2 score, suggesting the immediate past value is a significant predictor. Similar to LR, the XGB model with a lagged feature performed well, indicating the effectiveness of tree-based models on this dataset with this feature. Of the three standalone models LSTM was the best performer with low MAE and RMSE and higher r and R^2 values as can be seen from table 1.

Models which used STL decomposition and a single forecasting method to forecast the all the components separately and adding them to get the final forecast showed remarkable improvement in the performance than their standalone counterparts. Forecasting each component with separate LSTM models and combining them resulted in a notable improvement in performance compared to the standalone LSTM model, as indicated by lower MAE and RMSE, and higher Pearson's correlation coefficient r and R^2 scores. This suggests that modelling the components separately helped capture the underlying patterns more effectively. Using LR to forecast each component after STL decomposition also yielded a better performance than the standalone LR model, demonstrating the benefit of decomposing the time series. Of the three STL-LSTM, STL-XGB and STL-LR models, the STL-XGB model was the worst performer.

The hybrid approach, the STL-XGB-LR-LSTM model, utilizing different models for each component based on their characteristics, resulted in the best performance among all evaluated models, achieving the lowest MAE and RMSE, and the highest Pearson correlation and R2 scores. This suggests that tailoring the forecasting model to the specific nature of each component (e.g., using a model good at capturing non-linearities for the residual) can lead to improved overall forecasting accuracy. The graph of the predicted values vs the true values is shown in Figure 3. The plot of true vs. predicted values visually shows that there is a strong linearity between the true values and the forecasted values indicating good model performance. The dotted red line in the graph shows the perfect forecast.

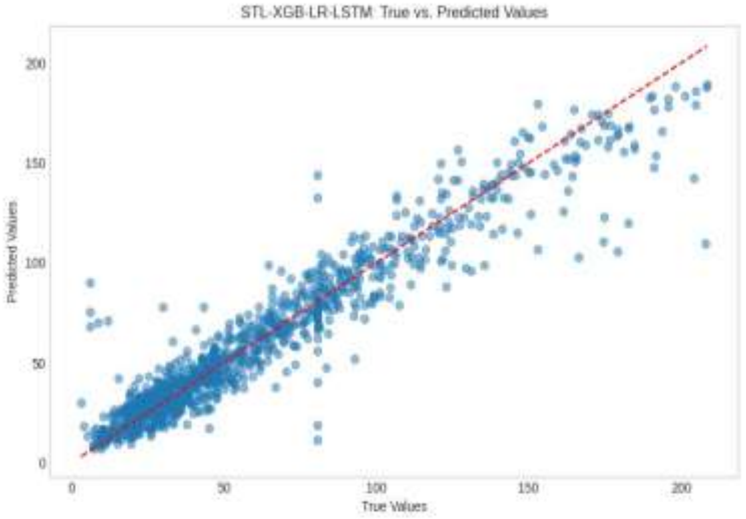

Figure 3: STL-XGB-LR-LSTM model's Forecasted vs True values plot

5. CONCLUSION

Based on the evaluation metrics, the hybrid forecasting approach combining STL decomposition with LR for the trend, XGB for the seasonal, and LSTM for the residual component provided the most accurate forecasts for the $PM_{2.5}$

dataset of Talkatora. Decomposing the time series into its constituent parts and applying models suited to the characteristics of each component proved to be a highly effective strategy.

6. REFERENCES

[1] Logothetis, S., Kosmopoulos, G., Panagopoulos, O., Salamalikis, V., & Kazantzidis, A. (2024). Forecasting the exceedances of $PM_{2.5}$ in an urban area. Atmosphere, 15(5), 594. https://doi.org/10.3390/atmos15050594

[2] Hassan, M. S., Bhuiyan, M. a. H., & Rahman, M. T. (2023). Sources, pattern, and possible health impacts of $PM_{2.5}$ in the central region of Bangladesh using PMF, SOM, and machine learning techniques. Case Studies in Chemical and Environmental Engineering, 8, 100366. https://doi.org/10.1016/j.cscee.2023.100366

[3] Sangkham, S., Phairuang, W., Sherchan, S. P., Pansakun, N., Munkong, N., Sarndhong, K., Islam, M. A., & Sakunkoo, P. (2024). An update on adverse health effects from exposure to $PM_{2.5}$. Environmental Advances, 100603. https://doi.org/10.1016/j.envadv.2024.100603

[4] Mahiyuddin, W. R. W., Ismail, R., Sham, N. M., Ahmad, N. I., & Hassan, N. M. N. N. (2023). Cardiovascular and respiratory health effects of fine particulate matters ($PM_{2.5}$): A review on Time series studies. Atmosphere, 14(5), 856. https://doi.org/10.3390/atmos14050856

[5] Deters, J. K., Zalakeviciute, R., Gonzalez, M., & Rybarczyk, Y. (2017). Modeling $PM_{2.5}$ urban pollution using machine learning and selected meteorological parameters. Journal of Electrical and Computer Engineering, 2017, 1–14. https://doi.org/10.1155/2017/5106045

[6] Doreswamy, N., S, H. K., Km, Y., & Gad, I. (2020). Forecasting air pollution particulate matter ($PM_{2.5}$) using machine learning regression models. Procedia Computer Science, 171, 2057–2066. https://doi.org/10.1016/j.procs.2020.04.221

[7] Putra, A. M. M. B., Martarizal, N., & Putra, R. M. (2020). Prediction of $PM_{2.5}$ and PM_{10} parameters using artificial neural network: a case study in Kemayoran, Jakarta. Journal of Physics Conference Series, 1528(1), 012036. https://doi.org/10.1088/1742-6596/1528/1/012036

[8] Chae, S., Shin, J., Kwon, S., Lee, S., Kang, S., & Lee, D. (2021). PM_{10} and $PM_{2.5}$ real-time prediction models using an interpolated convolutional neural network. Scientific Reports, 11(1). https://doi.org/10.1038/s41598-021-91253-9

[9] Zhou, Q., Jiang, H., Wang, J., & Zhou, J. (2014). A hybrid model for $PM_{2.5}$ forecasting based on ensemble empirical mode decomposition and a general regression neural network. The Science of the Total Environment, 496, 264–274. https://doi.org/10.1016/j.scitotenv.2014.07.051

[10] Niu, M., Gan, K., Sun, S., & Li, F. (2017). Application of decomposition-ensemble learning paradigm with phase space reconstruction for day-ahead $PM_{2.5}$ concentration forecasting. Journal of Environmental Management, 196, 110–118. https://doi.org/10.1016/j.jenvman.2017.02.071

[11] Jia, D., Ruan, W., Ma, R., Zhao, S., Wang, Y., Xu, W., Zhou, W., Ge, X., & Xu, Z. (2025). Hybrid framework for improved $PM_{2.5}$ prediction based on seasonal-trend decomposition and tailored component processing. Scientific Reports, 15(1). https://doi.org/10.1038/s41598-025-04597-x

[12] Sreenivasulu, T., & Rayalu, G. M. (2024). Enhanced $PM_{2.5}$ prediction in Delhi using a novel optimized STL-CNN-BILSTM-AM hybrid model. Asian Journal of Atmospheric Environment, 18(1). https://doi.org/10.1007/s44273-024-00048-7

[13] Wu, C., Wang, R., Lu, S., Tian, J., Yin, L., Wang, L., & Zheng, W. (2025). Time-Series Data-Driven $PM_{2.5}$ Forecasting: From theoretical framework to Empirical analysis. Atmosphere, 16(3), 292. https://doi.org/10.3390/atmos16030292

[14] Chen, T., & Guestrin, C. (2016). XGBoost. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). https://doi.org/10.1145/2939672.2939785

[15] Ghojogh, B., & Ghodsi, A. (2023). Recurrent Neural Networks and Long Short-Term Memory Networks: Tutorial and survey. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2304.11461

[16] Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. Physica D Nonlinear Phenomena, 404, 132306. https://doi.org/10.1016/j.physd.2019.132306

[17] Staudemeyer, R. C., & Morris, E. R. (2019). Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks. arXiv (Cornell University). https://doi.org/10.48550/arxiv.1909.09586