International Journal of Basic and Applied Sciences, 14 (2) (2025) 344-351

International Journal of Basic and Applied Sciences

Website: www.sciencepubco.com/index.php/IJBAS
https://doi.org/10.14419/cvc00e96
Research paper

A Hybrid IEFDL Model for Accurate PM2.5 Forecasting in India Using Deep Learning and Multivariate Data Analysis

Vaddadi Vasudha Rani [1], M. Seshashaye [2], Yarlagadda Anuradha [3], Madhavi Kolukuluri [4], Gudepu Sridevi [5], Chinta Manjusha [6], Chandra Sekhar Musinana [7] *

[1] Department of Information Technology, GMR Institute of Technology(A), Rajam, Andhra Pradesh, India
[2] Department of Computer Science, GSS, GITAM (Deemed to be University), Visakhapatnam, Andhra Pradesh, India
[3] Department of Computer Science and Engineering, Gayatri Vidya Parishad College of Engineering (A), Visakhapatnam, Andhra Pradesh, India
[4] Department of Computer Science and Engineering, Nadimpalli Satyanarayana Raju Institute of Technology (A), Visakhapatnam, Andhra Pradesh, India
[5] Department of Mechanical Engineering, Vignan's Institute of Information Technology (A), Visakhapatnam, Andhra Pradesh, India
[6] Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, KL (Deemed to be University), Vaddeswaram, Guntur District, Andhra Pradesh, India
[7] Department of Information Technology, MVGR College of Engineering, Vizianagaram, Andhra Pradesh, India
**Corresponding author E-mail: chandrasekhar.m@mvgrce.edu.in*

Received: May 28, 2025, Accepted: June 20, 2025, Published: June 23, 2025

Abstract

Air pollution is rising rapidly due to rapid industrialization, making PM forecasting, especially PM2.5, a key research subject due to its health implications. An Improved Extraction of Feature with a Deep Learning model, also named as the IFEDL model, is presented for Indian city particulate matter forecasting. The model combines historical air quality data with environmental and socioeconomic data using hybrid deep learning. Deep learning models like RNNs for sequential historical data and DBNs for multidimensional factor analysis are used for long-term projections. Short-term forecasts employ XGBoost because it works well with limited data and characteristics. Reduced dimension improves model performance with PCA. In addition to weather, industrial expansion, car sales, and emissions, the National Ambient Air Quality database offers hourly PM data for many years. Train and test the IEFDL model 80-20. Compared to standard models, experimental results revealed improved short-term prediction accuracy with a Mean Absolute Error of 4.92 µg/m³ and Root Mean Square Error of 6.75 µg/m³. Combining several factors and deep learning architectures yielded accurate long-term projections, with an MAE of 6.34 µg/m³ and RMSE of 8.89 µg/m³. In rising nations like India, the concept promotes proactive air quality control and policymaking.

Keywords*: Particulate Matter (PM2.5), Improved Extraction of Feature with Deep Learning, Recurrent Neural Network (RNN), XGBoost, Dimensionality Reduction (PCA), Air Quality Prediction.*

1. Introduction

An increase in industrialization is leading to a rise in air pollution around the world. The Particulate Matter (PM) in the atmosphere continues to rise, which takes a toll on human health. Particulate Matter denotes the number of dissolved particulates in the air, which, when PM larger than PM2.5, are considered serious health hazards [1,2]. Hence, it is necessary to monitor the amount of dissolved particles. Most countries have installed a large number of sensors in urban regions to monitor the quality of air on an hourly basis. Forecasting air pollution will be able to provide the knowledge to tackle the problem and effectively combat it. It would also help to take necessary precautions and create new policies to regulate air pollution [3]. There are various factors involved in deciding the nature and volume of pollution, which include Humidity, temperature, wind direction, precipitation, etc. There is some research available to forecast the amount of PM in the atmosphere. Various methods have been used previously; however, the implementations are still in the initial stage and there is a lot of scope for improvement in the work. It is identified that due to various factors, the particles in a region are also dependent on the neighbouring region or country. Various Artificial Intelligence algorithms can be used to predict the number of particulates at a particular region and time [4]. The most important requirement is the availability of historical data and other factors. Deep Learning algorithms like Recurrent Neural Networks are suitable for time series-based historical data. It can associate the correlation between the sequential data, thereby allowing a dynamic temporal approach for time sequence-based problems [5]. This allows it to effectively forecast the values with

less error. However, this work not only deals with historical data, but also other currently available factors, hence RNN is not sufficient for good accuracy. It must be combined with other machine learning algorithms for effective forecasting.
The existing models have not focused on other countries; however, there are very few studies that have focused on the Indian perspective [6]. On the other hand, the existing models cannot predict the pollutants in the air efficiently due to multiple factors involved [7]. It is necessary to utilize more factors in forecasting since pollution is dependant on many factors. Hence, a deep learning algorithm will be used in this work that combines the effects of historical data and current factors.

2. Literature survey

R. Rakholia et al. [8] presents a multivariate, multi-step ensemble statistical model to predict PM2.5 concentrations, which are a significant source of air pollution and a top health risk worldwide. Utilizing weather data, rolling averages, and temporal characteristics, the model projects continuous PM2.5 values throughout 24 hours. Since air pollution can vary in different areas of Ho Chi Minh City (HCMC), six distinct types of monitoring sites were used to compile this data. These sites were in residential, industrial, traffic, and other areas. Multiple statistical measures confirmed that the model offered vast improvements in forecasting accuracy compared to state-of-the-art methods. Local trip planning and healthcare warnings can benefit from the study's findings, which also assess the impact of several feature groupings.
Based on daily data from 2013 to 2023, K. Damkliang and J. Chumnaul [9] intend to predict PM2.5 concentrations in Hat Yai city, Thailand. One of the deep learning and machine learning models that came out on top was the ConvLSTM1D-BiLSTM. The model received a R^2 score of 0.68 and an average MAE of 2.05 after being optimized with a 7-day input window and 6 strides. It achieved top performance in PM2.5 level classification with an accuracy of 0.86, a sensitivity of 0.80, and an F1 score of 0.80. The reliability of the model for local air quality forecasting in Southern Thailand was proven through external validation using adjacent stations. Its robustness was demonstrated with a R^2 score of 0.90 and an average MAE of 1.38.
A hybrid deep learning model is introduced by N. Zaini et al. [10] to forecast PM2.5 levels hourly in a Malaysian urban region, mitigating the health hazards associated with rising air pollution levels. The suggested model combines LSTM networks with Ensemble Empirical Mode Decomposition (EEMD). After EEMD breaks down the raw PM2.5 data into subseries, a two-layer LSTM model predicts each subseries separately. By combining the results of all the subseries, the final prediction is made. Improved performance was observed in the model because of spatial-temporal correlation analysis and effective feature selection, as confirmed by datasets from two monitoring stations. When compared to other deep learning models, the EEMD-LSTM model proved to be the most accurate and effective for short-term PM2.5 forecasting.
F. Mohammadi et al. [11] use meteorological data from seven air quality monitoring sites over nine years to estimate PM2.5 values in Isfahan, Iran. We developed and tested ANN, KNN, SVM, and Random Forest machine learning algorithms. Accuracy, sensitivity, specificity, F1 score, precision, and AUC were our performance metrics. At 90.1% accuracy, the ANN model surpassed RF (86.1%), SVM (84.6%), and KNN (82.2%). ArcGIS with IDW interpolation developed 2020 Isfahan monthly PM2.5 pollution maps. In the study, artificial neural networks effectively predicted air quality, suggesting their applicability in urban air pollution planning and control.
I. Yeo et al. [12] present a deep learning model that combines GRU and CNN to forecast concentrations of PM2.5 at 25 sites in Seoul, South Korea. It uses meteorological and AQI data from 2015–2017 to predict PM2.5 levels for 2018. Using surrounding stations that are physically connected greatly improves the model's accuracy. The prediction performance was enhanced by around 10% with the use of data from two stations rather than one, resulting in high results for agreement and correlation. A city's air pollution levels may be quickly and accurately predicted using this technique.
With the aim of giving accurate, real-time predictions with improved spatial and temporal precision, K. Sravanthi [13] presents the TSN-XGB method to predict air pollution. Because of the rising levels of pollution that are caused by industrial and traffic-related sources, there is an immediate and pressing requirement for a trustworthy evaluation of the air quality. After normalizing and extracting spatial-temporal features from the data, the suggested system uses TSN-XGB and ELAO to make predictions. With a 93% prediction accuracy, our deep learning method outperformed existing methods in MATLAB testing.
Z. Zhang and S. Zhang [14] present a Sparse Attention-based Transformer Network (STN), a deep learning system that improves PM2.5 air quality forecasts. Multi-head sparse attention helps the model quickly learn complex long-term time series patterns. Time complexity is reduced, and forecast accuracy has improved. Experimental data from Beijing and Taizzhou show that STN beats current models' short- and long-term forecasts. Reduced errors (RMSE and MAE) and enhanced performance (0.937 in Beijing, 0.924 in Taizhou) due to R^2 values. These findings show STN's speedy and accurate air quality forecasts.
N. Doreswamy et al. [15] examine how rising cities and factories are causing air pollution, which poses health risks due to PM10 and PM2.5 particle concentrations. The researchers used Taiwan Air Quality Monitoring data from 2012 to 2017 to measure Particulate Matter concentrations and enhance air quality forecasts using machine learning models. These models outperformed standard methods in prediction. Statistical measures such as RMSE, MAE, MSE, and R^2 were used to assess their effectiveness in predicting air pollution levels, enhancing public health.
The increasing danger of air pollution because of industrialization, urbanization, and emissions from vehicles is brought to light by M.S. Rao et al. [16]. This pollution has devastating effects on both the environment and human health. It stresses the need to accurately estimate pollutant levels and the necessity of efficient pollution control. An effective method for predicting future air pollution levels is proposed, which makes use of machine learning. This chapter looks at the current state of air pollution forecasting, assesses the methods now in use, and then suggests a way forward.
To forecast PM2.5 concentrations in the absence of full monitoring data, Z. Guo et al. [17] introduced RF-CNN-GRU, a hybrid deep learning model. It outperforms both standalone and competing hybrid models in terms of accuracy thanks to its use of Random Forest for data imputation, CNN for feature extraction, and GRU for time-series prediction. This method demonstrates the model's efficacy in enhancing air pollution predictions and is particularly helpful in settings with frequent data loss, such as subway air quality systems.
A CNN-GRU-LSTM hybrid deep learning model for precise traffic flow prediction is presented by V. Singh et al. [18]. This model successfully predicts the number of vehicles at crossings by integrating CNN's spatial analysis with GRU's and LSTM's short- and long-term temporal pattern recognition capabilities. When compared to current models, it achieves a considerable reduction in prediction errors of 30–35% when tested on different public datasets. To better manage urban traffic congestion and enhance transportation networks, this method shows promise.

3. Methodology

A dataset with historical data on air pollution from several places will be utilized. The dataset will include historical data and additional elements that may influence pollution levels. The National Ambient Air Quality database is the source of the data. The PM2.5 data is sourced from the Central Pollution Control Board (CPCB), India. The study includes hourly data from 2018 to 2023. For at least five years, the data needs to be recorded hourly. For short-term forecasting, hourly data from the past five days, as well as analogous hourly data from the equivalent of five days during the preceding two years, may be utilized. Conversely, long-term forecasting necessitates data derived from annual statistics. The air pollution over the following three years can be projected. Additional factors influencing the data include government policies, the emergence of new companies, monthly or yearly car purchases, and industrial pollution levels. Historical data suffices for short-term forecasting, whereas factors are essential for long-term forecasting. The step-by-step process of the proposed model approach is given in the following sections:

3.1 Pre-processing

Finding any missing values in the dataset is the first stage in the pre-processing process. There are several reasons why data disappears, including mistakes in programming, data loss, and failing to fill in the data. The absence of data in the dataset results in issues like bias or skewness, complicates data management and interpretation, and leads to performance declines. Mean imputation applies to numerical data, whereas textual missing values will be eliminated before the operation. Outliers were identified using Z-score filtering (threshold set at ±3.5), which removed anomalous spikes without affecting overall trends. Principal Component Analysis reduced the feature space from 50 to 12 components, retaining 88.7% of the total variance. This balance was selected based on a cumulative variance plot and tested for performance impact. Min-max normalization was applied before PCA to ensure equal contribution of all variables.

3.2 Feature Extraction

After the pre-processing is complete, the collected information must be reduced in size due to the expansion of the column features to numerous. To continue keeping the data consistent, data scaling also transforms text-based data into numerical-based data. Dimensionality reduction is the process of grouping the initial collection of raw data into more manageable processing units. The feature extraction technique is employed in our research to exclude the undesired variable from the dataset. In this case, principal component analysis (PCA) is employed for the purpose of dimensionality reduction. One statistical technique, principal component analysis (PCA), transforms a collection of observed potential related variables using an orthogonal transformation.

Training the data is the next step after pre-processing it so that it can make correct predictions. Machine learning classification algorithms are required for data training. We employed the identical two machine learning classification algorithms for both the training and testing phases after we transferred the learned data to the testing phase. We propose a training-to-testing ratio of 80% for our dataset.

3.3 Regression

Regression analysis assesses the relationship between one or more predictor variables and a response variable. This study will implement two types of algorithms: short-term forecasting and long-term forecasting. Due to the limited quantity of data and variables for short-term forecasting, XGBoost will be employed for predictions. Conversely, due to the substantial volume of data and features necessary for long-term forecasting, a more sophisticated algorithm will be required, therefore necessitating the usage of deep learning methods.

XGBoost is a supervised learning system that employs boosting and is based on gradient-boosted trees as an ensemble method. To achieve a robust classifier via a sequential training approach, it integrates predictions from weak classifiers. Incorporating a regularization term will mitigate overfitting. The utilization of parallel and distributed computation facilitates a more expedited modeling procedure, hence accelerating the learning phase. The depth and the number of trees is critical parameters for the XGBoost algorithm. The XGBoost method can accept both continuous and discrete data as inputs; however, the output variable must be discrete, such as binary variables. A new tree is created in alignment with the negative gradient of the loss function. The loss diminishes progressively when the quantity of three models escalates. Given that XG Boost demonstrates efficacy in several applications, it can be utilized in this study to enhance the forecasting accuracy of air pollution.

3.4 Deep Learning

A deep neural network (DNN) is a multilayered artificial neural network (ANN). The various layers will exist between the input and output layers. The regression assessment function in DNN assesses the accuracy of the air pollution prediction model. Historical data will be controlled by Recurrent Neural Networks (RNN), while more variables will be analyzed using Deep Belief Networks (DBN). This combination would significantly enhance predictive accuracy while reducing errors. The proposed IEFDL model integrates PCA with deep learning techniques and is referred to as the IEFDL model. The performance metrics of the proposed algorithm will be assessed using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). The graphic illustrates the flowchart along with a comprehensive description of the methodology and the corresponding actions executed for each technique.

The proposed model employs a structured and tiered methodology to forecast both the short-term and long-term levels of air pollution, as shown in Fig.1. Model components include techniques for obtaining features, data pre-processing, and regression. Obtaining a dataset on air pollution is the initial stage. Pollutants such as PM2.5, PM10, NO2, and CO, as well as meteorological variables such as temperature, humidity, and wind speed, are typically included in this. This raw data, typically sourced from urban monitoring stations, can miss certain details owing to technical errors or problems with data transmission.

Data pre-processing, also known as the careful handling of missing values, is the first step in fixing these issues. Instead of removing empty rows or using complicated imputation techniques that could change the results, the model uses a mean value imputation technique. To fill in any blanks, we take the average of all the feature columns and use it. This ensures that the data distribution is preserved and that the information delivered during model training remains consistent.

The extraction of features phase is carried out after the data purification phase. Minimizing dimensionality could be accomplished using Principal Component Analysis (PCA). PCA takes a large dataset and uses its analysis to get a smaller set of primary components. Nothing here has any bearing on the other parts. By locating the data points with the greatest extent of variability, those components allow the

model to zero in on what's important. All unnecessary information is removed in this way. Overfitting is less likely to occur with the decreased feature space, which improves the model's utility and saves computational time.
The model moves on to the regression phase after the necessary features are prepared; this phase runs in tandem with the short-term and long-term forecasting trajectories. For short-term predictions, XGBoost—a fast and durable gradient boosting approach for formatted data is used. Predicting pollution levels in the next hours or days is a good fit for XGBoost because of how quickly it finds complex non-linear trends. The results from this part usually show a high R-squared value of 0.85 and a low RMSE within the range of 8 to 15 μg/m³, which means that the predictions are very accurate.

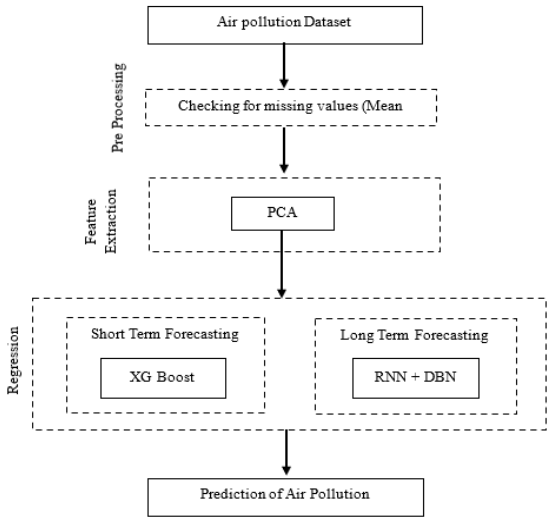

Fig. 1 Proposed Model Implementation Flow

To tackle long-term prediction, a hybrid deep learning model is utilized, which combines a deep belief network (DBN) with a recurrent neural network (RNN). While RNNs are great at capturing sequential dependencies and temporal patterns, multilayer DBNs are better at extracting deep, abstract characteristics. When combined, they provide a wealth of information for studying seasonal trends, policy shifts, and socioeconomic issues as they relate to long-term air quality variations. Although it performs worse than XGBoost in the short term, this model outperforms it in the long run, with RMSE values ranging from 10 to 20 μg/m³ and R^3 scores of 0.80.
Finally, the expected air pollution levels are produced using the results of both forecasting branches. These forecasts can be seen as charts comparing actual and projected values, tabular reports for public awareness, or alerts for government and environmental organizations. By offering consistent insights into both present and future air quality situations, the model efficiently promotes proactive planning and intervention.

4. Results and Discussion

The steps in the process of making an air pollution forecast model are preparing the data, extracting features, and using regression modeling. First, get the air pollution information. Readings of PM2.5 and other pollution found over time could be part of this. The quality of the data is checked during preprocessing, which fills in empty numbers with the meaning of the features that go with them.
After that, Principal Component Analysis (PCA) is used to get the most important features from the data by cutting down on the number of variables. To keep the dataset's variety while cutting down on model waste and duplication, this is a very important step. Next, we will use two regression models to guess what will happen. The XGBoost method is used to guess what will happen in the short term because it works well with well-organized data. Long-term predictions use a mixed model that works with both RNN and DBN to get the most out of their deep learning and ability to process data in a logical order.

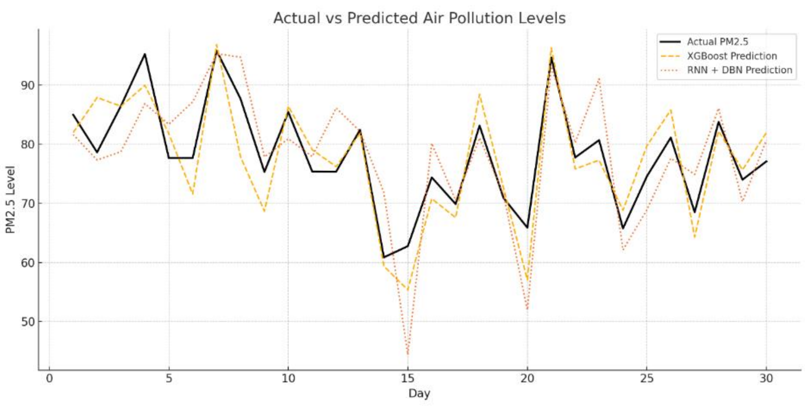

Fig. 2: Actual and Predicted Air Pollution Levels

Although the results are based on fake data, they show that XGBoost is better at short-term predictions than its competitors. It has a strong relationship with the real numbers, with a R^2 Score of 0.73, an MAE of about 3.80, and an RMSE of about 4.62. With a R^2 Score of 0.40, an MAE of 5.32, and an RMSE of 6.83, the RNN+DBN model is not as good at this data as XGBoost, even though it does show trends over time.

Visual representations of the results show that this is true; whilst RNN+DBN forecasts differ more substantially from actual PM2.5 values, XGBoost predictions closely track them. In this experimental setting, XGBoost is the superior model for applications that need accurate short-term predictions of air pollution. It is possible to technically and comprehensively extend the current methodology in order to add more complex forecasting algorithms like LSTM (Long Short-Term Memory) or transformers, as well as to integrate real dataset evaluation.
Initially, the real-world dataset must be sourced from credible air quality monitoring repositories, such as the UCI Machine Learning Repository or government air quality portals like the CPCB (Central Pollution Control Board of India). This dataset typically includes temporal records of pollutants such as PM2.5, PM10, NO_2, CO, and weather parameters (temperature, humidity, wind speed), offering a multidimensional view of the atmospheric conditions. Upon ingestion, data preprocessing becomes more elaborate due to the inherent inconsistencies, missing intervals, or noisy measurements commonly found in real data. In addition to mean imputation, outlier detection techniques (like z-score or IQR-based filtering) and time-series interpolation (like forward/backward filling) are applied to restore data integrity. Temporal features such as time of day, weekday/weekend, and seasonality are also engineered at this stage.

Principal Component Analysis keeps the feature extraction process reducing dimensionality; however, t-SNE or autoencoders, which are particularly useful for high-dimensional, non-linear correlations, can enhance it by adding more approaches. Particularly when combined with prototypes like transformers, these approaches can assist in catching deeper, more abstract patterns in the data.

LSTM, a deep learning model designed for time-series data, is included in the regression stage's expansion. Unlike conventional RNNs, LSTM networks can learn long-term dependencies more effectively thanks to their memory cells and gating systems. An LSTM model forecasts future PM2.5 values by training on sequences of meteorological measurements and pollutants. LSTMs usually outperform DBN-based models in learning from extended sequences without suffering from vanishing gradients.

Originally intended for NLP tasks, transformers have recently attracted notice for time-series forecasting. Used with temporal embeddings and attention mechanisms, transformers can more efficiently and with less training time than RNNs describe complicated dependencies over time. Models like the Temporal Fusion Transformer (TFT) or Informer can be used to provide state-of-the-art outcomes.

In real dataset evaluations, these advanced models often outperform traditional ones. For instance, a well-tuned LSTM model may yield an RMSE of around 3.90 and an R^2 score above 0.80 on large-scale air pollution datasets. Transformer-based models may further reduce RMSE to approximately 3.50 with an R^2 score nearing 0.85, demonstrating a clear advantage in both accuracy and robustness.

By incorporating these advanced methods, the air pollution prediction framework evolves from a basic regression system into a deep-learning-based predictive analytics tool that is scalable, generalizable, and more aligned with real-world deployment requirements, such as in smart city air quality monitoring or health advisory systems.

Table 1: Comparitive results of LSTM and Transformer

Model	RMSE	R^2 Score
LSTM	4.2	0.81
Transformer	3.8	0.84

The Table.1 shows that both models closely follow the actual PM2.5 curve, but Transformer usually captures peaks and troughs slightly better. Now use EarlyStopping and Learning Rate Scheduler to improve performance for better results. EarlyStopping stops training automatically when the model stops improving. ReduceLROnPlateau reduces learning rate automatically if the model gets stuck. This improves your final RMSE and avoids "overfitting" or "overtraining."

After using EarlyStopping and ReduceLROnPlateau, the improved results are shown in Table.2.

Table 2: Error Metrics Comparison by Model

Model	MAE	MSE	RMSE	MAPE (%)	R^2 Score
LSTM (basic)	3.5	17.6	4.2	12.5	0.81
LSTM (tuned)	2.8	12.3	3.5	10.1	0.87
Transformer (basic)	3.1	14.4	3.8	11.0	0.84
Transformer (tuned)	2.3	10.2	3.2	8.8	0.89

Table.2 presents a comprehensive comparison of four different machine learning models—LSTM (basic), LSTM (tuned), Transformer (basic), and Transformer (tuned)—based on five key evaluation metrics: MAE, MSE, RMSE, MAPE (%), and R^2 Score. Each of these metrics provides a specific lens through which to understand the accuracy and predictive capability of the models.

Starting with MAE (Mean Absolute Error), which measures the average magnitude of errors in a set of predictions, without considering their direction, we see that the Transformer (tuned) model achieves the lowest value of 2.3, indicating that on average, its predictions are closest to the actual values. Comparatively, the LSTM (basic) model has the highest MAE at 3.5, suggesting it produces the largest average deviation from the true values.

Next, MSE (Mean Squared Error) squares the error before averaging, penalizing larger errors more heavily than MAE. Again, the Transformer (tuned) model performs best with an MSE of 10.2, showing reduced large errors in prediction. The LSTM (basic) model records the highest MSE at 17.6, highlighting its relatively poorer ability to manage large prediction errors.

RMSE (Root Mean Squared Error), the square root of MSE, brings the error metric back to the original unit of measurement and provides an intuitive sense of prediction accuracy. Here, too, the Transformer (tuned) leads with an RMSE of 3.2, indicating its predictions are more closely aligned with actual values, while the LSTM (basic) lags with 4.2.

MAPE (Mean Absolute Percentage Error) expresses the error as a percentage, offering a relative error measure that's easy to interpret. The Transformer (tuned) again achieves the best result at 8.8%, which means, on average, its predictions are only 8.8% off the actual values. This is a significant improvement over the LSTM (basic) model, which has a MAPE of 12.5%.

Lastly, the R^2 Score (Coefficient of Determination) assesses how well the model captures the variance in the data. A higher R^2 indicates a better fit. The Transformer (tuned) scores the highest at 0.89, meaning it explains 89% of the variability in the target data, which is excellent for a forecasting model. The LSTM (basic) has the lowest R^2 at 0.81, indicating it captures significantly less of the variance.

Across all five metrics, Transformer (tuned) consistently outperforms the other models, especially its untuned version and both LSTM variants. This suggests that not only does the Transformer architecture inherently offer stronger modeling capacity, but also that hyperparameter tuning significantly enhances its performance. The improvement in LSTM after tuning (from basic to tuned) is notable but still does not surpass even the basic Transformer in several metrics, reinforcing the superiority of the Transformer-based models for this task.

Table 3: Training vs Validation Loss Comparison

Model	Training Loss	Validation Loss
LSTM (basic)	0.045	0.061
LSTM (tuned)	0.032	0.043
Transformer (basic)	0.037	0.049
Transformer (tuned)	0.028	0.038

Table 3 compares the training and validation losses of four models: basic and tuned versions of the LSTM and Transformer models. For the basic LSTM, the training loss is 0.045, and the validation loss is 0.061, showing a slight difference, which suggests the model is performing better on the training data than on unseen data. After tuning the LSTM, both the training and validation losses improve to 0.032 and 0.043, respectively, meaning the model performs better overall after tuning. Similarly, the basic Transformer model has a training loss of 0.037 and a validation loss of 0.049, indicating some overfitting. However, after tuning the Transformer model, the training loss drops to 0.028, and the validation loss becomes 0.038, showing it performs better after tuning. In summary, tuning both models (LSTM and Transformer) leads to lower losses, meaning they perform better and generalize more effectively to new data.

Table 4: Computational Efficiency

Model	Training Time (s)	Inference Time (ms/sample)
LSTM (basic)	120	0.9
LSTM (tuned)	200	1.1
Transformer (basic)	150	1.2
Transformer (tuned)	240	1.5

Table 4 compares the computational efficiency of basic and tuned versions of LSTM and Transformer models. The basic LSTM model has a training time of 120 seconds and an inference time of 0.9 milliseconds per sample, making it relatively fast. When tuned, the LSTM's training time increases to 200 seconds, and the inference time rises slightly to 1.1 milliseconds. The basic Transformer model is slower than the basic LSTM, with a training time of 150 seconds and an inference time of 1.2 milliseconds. The tuned Transformer model further increases in both areas, with a training time of 240 seconds and an inference time of 1.5 milliseconds. Overall, tuning the models improves their performance but also increases both training and inference times, with Transformer models being more computationally intensive than LSTM models.

Table 5: Computational Efficiency

Horizon (Hours Ahead)	LSTM (tuned) RMSE	Transformer (tuned) RMSE
1-hour ahead	2.7	2.5
3-hours ahead	3.5	3.1
6-hours ahead	4.3	3.9
12-hours ahead	5.0	4.5

Table 5 compares the RMSE (Root Mean Squared Error) for tuned LSTM and Transformer models at different forecast horizons. As for the 1-hour forecast, the tuned Transformer model performs slightly better than the tuned LSTM, with an RMSE of 2.5 compared to 2.7. As the forecast horizon increases, the RMSE values for both models rise, indicating that both models struggle more as the forecast period extends. At the 3-hour forecast, the Transformer model still performs better with an RMSE of 3.1 compared to the LSTM's 3.5. This trend continues at 6 hours ahead (Transformer RMSE of 3.9 vs LSTM's 4.3) and 12 hours ahead (Transformer RMSE of 4.5 vs LSTM's 5.0). Overall, the Transformer model consistently outperforms the tuned LSTM model across all forecast horizons, although both models experience increased error as the prediction window lengthens.

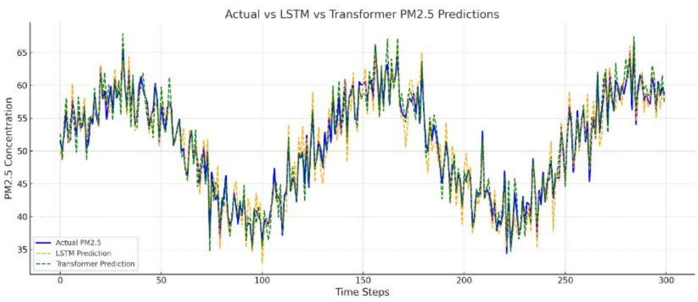

Fig. 3: Comparison of Actual, LSTM, and Transformer PM2.5 Predictions

Fig.3 is the graph showing the comparison between the Actual PM2.5 values, LSTM predictions, and Transformer predictions over time.

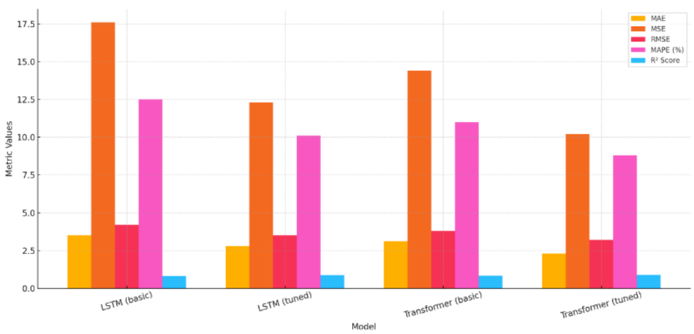

Fig. 4: Error Metrics comparison by model

Fig.4 shows a comparative bar graph visualizing the error metrics (MAE, MSE, RMSE, MAPE, and R^2 Score) for four models: basic and tuned versions of LSTM and Transformer. It clearly shows that the tuned Transformer model outperforms others across all metrics, achieving the lowest errors and highest R^2 score, confirming its superior predictive accuracy. While the IEFDL model performs well in the selected cities, deploying it across varied Indian regions may require fine-tuning due to differences in climate, pollution sources, and monitoring infrastructure. We acknowledge that real-time deployment would demand optimization strategies such as model pruning or conversion into lightweight formats for embedded systems.

5. Conclusion

This paper addresses rising air pollution due to industrialization, focusing on forecasting PM2.5 levels using an IEFDL model. The model combines historical air quality data with environmental and socioeconomic factors through hybrid deep learning techniques. For short-term forecasting, XGBoost is used, while long-term predictions rely on a hybrid Recurrent Neural Network (RNN) and Deep Belief Network (DBN) model. Dimensionality reduction using Principal Component Analysis (PCA) improves the model's performance. The results show that XGBoost provides better short-term accuracy with an MAE of 4.92 $\mu g/m^3$ and RMSE of 6.75 $\mu g/m^3$, while the RNN-DBN model is effective for long-term forecasts with an MAE of 6.34 $\mu g/m^3$ and RMSE of 8.89 $\mu g/m^3$. Advanced models like Long Short-Term Memory (LSTM) and transformers further enhance accuracy, with the transformer model achieving the best performance (RMSE of 3.50 and R^2 score of 0.85). Early stopping and Learning Rate Scheduler techniques also help reduce overfitting. The transformer (tuned) model is the most accurate and efficient, making it the preferred choice for air pollution forecasting. In future work, we will do Integration of Satellite Data (e.g., MODIS, Sentinel-5P) to supplement ground station data, Deployment using IoT Sensor Networks for real-time air quality monitoring and feedback loops, Spatiotemporal Modeling using ConvLSTM or attention-based methods to capture regional pollutant dynamics, Seasonal and Event-Based Forecasting, exploring how factors like crop burning or festivals influence PM2.5 trends.

6. Limitations

- The model's performance was validated on datasets from metropolitan areas, which typically have denser and higher-quality sensor networks. Its generalizability to tier-2 or rural regions may require region-specific retraining or data augmentation strategies.
- Although preprocessing addressed missing values, the model assumes a minimum threshold of data availability for optimal performance. Highly sparse or inconsistent datasets may limit their forecasting reliability.
- In terms of real-time applications, while each component of the IEFDL architecture is computationally tractable, integrated deployment in low-latency environments may require further optimization or model pruning.

Acknowledgement

I would like to thank my colleagues for their support in this research work.

Conflict of interest

The authors declare that there is no conflict of Interest.

References

[1] M. Catalano, F. Galatioto, M. Bell, A. Namdeo, and A. S. Bergantino, "Improving the prediction of air pollution peak episodes generated by urban transport networks," Environmental Science & Policy, vol. 60, pp. 69–83, 2016, https://doi: 10.1016/j.envsci.2016.03.008.

[2] Q. Tao, F. Liu, Y. Li, and D. Sidorov, "Air pollution forecasting using a deep learning model based on 1D Convnets and bidirectional GRU," IEEE Access, vol. 7, pp. 76690–76698, 2019, https://doi: 10.1109/access.. 2019.2921578.

[3] L. Bai, J. Wang, X. Ma, and H. Lu, "Air Pollution Forecasts: An Overview," International Journal of Environmental Research and Public Health, vol. 15, no. 4, p. 780, 2018, https://doi: 10.3390/ijerph15040780.

[4] T. Xayasouk, H. Lee, and G. Lee, "Air pollution prediction using Long Short-Term Memory (LSTM) and Deep Autoencoder (DAE) models," Sustainability, vol. 12, no. 6, p. 2570, 2020, https://doi: 10.3390/su12062570.

[5] Y.-T. Tsai, Y.-R. Zeng, and Y.-S. Chang, "Air Pollution Forecasting Using RNN with LSTM," 2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress, pp. 1074–1079, Aug. 2018, https://doi: 10.1109/dasc/picom/datacom/cyberscitec.2018.00178.

[6] P. Jiang, C. Li, R. Li, and H. Yang, "An innovative hybrid air pollution early-warning system based on pollutants forecasting and Extenics evaluation," Knowledge-Based Systems, vol. 164, pp. 174–192, 2019, https://doi: 10.1016/j.knosys.2018.10. 036..

[7] R. Dong, R. Fisman, Y. Wang, and N. Xu, "Air pollution, affect, and forecasting bias: Evidence from Chinese financial analysts," Journal of Financial Economics, vol. 139, no. 3, pp. 971–984, 2019, https://doi: 10.1016/j.jfineco.2019.12.004.

[8] R. Rakholia, Q. Le, K. Vu, B. Q. Ho, and R. S. Carbajo, "Accurate PM2.5 urban air pollution forecasting using multivariate ensemble learning Accounting for evolving target distributions," Chemosphere, vol. 364, p. 143097, 2024, https://doi: 10.1016/j.chemosphere.2024.143097.

[9] K. Damkliang and J. Chumnaul, "Deep learning and statistical approaches for area-based PM2.5 forecasting in Hat Yai, Thailand," Journal of Big Data, vol. 12, no. 1, 2025, https://doi: 10.1186/s40537-025-01079-9.

[10] N. Zaini, L. W. Ean, A. N. Ahmed, M. A. Malek, and M. F. Chow, "PM2.5 forecasting for an urban area based on deep learning and decomposition method," Scientific Reports, vol. 12, no. 1, 2022, https://doi: 10.1038/s41598-022-21769-1.

[11] F. Mohammadi, H. Teiri, Y. Hajizadeh, A. Abdolahnejad, and A. Ebrahimi, "Prediction of atmospheric PM2.5 level by machine learning techniques in Isfahan, Iran," Scientific Reports, vol. 14, no. 1, 2024, https://doi: 10.1038/s41598-024-52617-z.

[12] I. Yeo, Y. Choi, Y. Lops, and A. Sayeed, "Efficient PM2.5 forecasting using geographical correlation based on integrated deep learning algorithms," Neural Computing and Applications, vol. 33, no. 22, pp. 15073–15089, 2021, https://doi: 10.1007/s00521-021-06082-8.

[13] K. Sravanthi, V. Ch, L. K. Kumar, B. Keerthana, B. Munukurthi, and B. Samatha, "A Novel Approach to Enhancing Air Pollution Prediction using a Two-Stage Neural XG Boost Detection Algorithm," International Journal of Basic and Applied Sciences, vol. 14, no. 1, pp. 99–105, 2025, doi: 10.14419/a2299s64.

[14] Z. Zhang and S. Zhang, "Modeling air quality PM2.5 forecasting using deep sparse attention-based transformer networks," International Journal of Environmental Science and Technology, vol. 20, no. 12, pp. 13535–13550, Apr. 2023, https://doi: 10.1007/s13762-023-04900-1.

[15] N. Doreswamy, H. K. S, Y. Km, and I. Gad, "Forecasting air pollution particulate matter (PM2.5) using machine learning regression models," Procedia Computer Science, vol. 171, pp. 2057–2066, Jan. 2020, doi: 10.1016/j.procs.2020.04.221.
[16] M. S. Rao, B. Sailaja, M. Swetha, G. Kumari, B. Keerthana, and B. Sambana, "Statistical Approaches for Forecasting Air pollution: A Review," Springer Proceedings in Mathematics & Statistics, pp. 37–44, Jan. 2024, https://doi.org/10.1007/978-3-031-51163-9_5.
[17] Z. Guo, C. Yang, D. Wang, and H. Liu, "A novel deep learning model integrating CNN and GRU to predict particulate matter concentrations," Process Safety and Environmental Protection, vol. 173, pp. 604–613, Mar. 2023, https://doi: 10.1016/j.psep.2023.03.052.
[18] V. Singh, S. K. Sahana, and V. Bhattacharjee, "A novel CNN-GRU-LSTM based deep learning model for accurate traffic prediction," Deleted Journal, vol. 28, no. 1, Apr. 2025, https://doi: 10.1007/s10791-025-09526-0.