Journal of Trends in Computer Science and Smart Technology (ISSN: 2582-4104)
www.irojournals.com/tcsst/

A Multifaceted Framework for Predicting Ambient Air Pollution in the National Capital Region

Dhanalakshmi Subramanian[1], Poongothai Marimuthu[2], Nithyalakshmi Ramadoss[3]

Department of Electronics and Communication Engineering, Coimbatore Institute of Technology, Coimbatore, India.

E-mail: [1]dhanalakshmi.s@cit.edu.in, [2]poongothai@cit.edu.in, [3]nithyalakshmi.r@cit.edu.in

Orcid ID: [1]0000-0002-0671-1001, [2]0000- 0002-9203-5256, [3]0000-0002-0450-4615

Abstract

Air quality forecasting models anticipate and control pollution concentrations quickly. In this research, two diversified datasets are analyzed and modeled using the proposed novel Hybrid Forecasting Model (HFM). PM10, PM2.5, NO, and NO2 pollutant concentrations for the National Capital Region, Delhi, were collected from the Central Pollution Control Board as Dataset 1. A Raspberry Pi hardware integrated with an MCP3008 Analog-to-Digital Converter and pollutant sensors like MQ135 and GP2Y1010AU0F was assembled to collect PM2.5, CO, and NH3 pollutant concentrations at the study site as Dataset 2. The results obtained for the proposed HFM are compared with three existing algorithms in terms of R-squared and Mean Squared Log Error. The novelty of this work is to test the performance of the algorithm by executing it on a computer with an Intel 8265U processor and also on a Raspberry Pi integrated with an MCP3008 Analog-to-Digital Converter. The experimental results indicate that the Raspberry Pi requires approximately 3.37% of the power consumed by the Intel 8265U processor. Conversely, the Intel 8265U processor achieves approximately 86.67% lower latency, thereby delivering significantly faster computational performance compared to the Raspberry Pi.

Keywords: Long Short-Term Memory, Convolutional Neural Network, Auto Regressive Integrated Moving Average, R Squared, Mean Squared Log Error, National Capital Region.

1. Introduction

According to the fifth annual World Air Quality Report 2022, India ranked eighth among the most polluted countries, with an annual average PM2.5 concentration of 53.3 micrograms per cubic meter, which is more than 10 times the WHO's recommended levels [1]. With an annual average PM2.5 concentration of 92.6 micrograms per cubic meter, Delhi ranked as the second most polluted capital city globally. Similarly, NO and NO2 are released from the combustion of fuels in the transportation and industrial sectors. In India, forecasting air pollutant levels in advance becomes absolutely essential due to industrialization [2]. Leading approaches in time series forecasting include recurrent neural networks, artificial neural networks, and the statistical concept of autocorrelation. Neural network concepts are dominant

DOI: https://doi.org/10.36548/jtcsst.2026.3.007
Received: 02.05.2026, received in revised form: 04.06.2026, accepted: 28.06.2026, published: 09.07.2026

tools for learning from data and creating estimates based on patterns and trends, while statistical time series forecasting models are also essential tools for forecasting future events based on past time-stamped data points [3].

Thus, this work proposes a Hybrid Forecasting Model (HFM), which combines neural networks and statistical time series forecasting models. This model is used to predict the future concentrations of PM10, PM2.5, NO, and NO2 in the National Capital Region (NCR) Delhi (dataset_1) and to predict the future concentrations of PM2.5, CO, and NH3 in the on-site area (dataset_2).

The main objectives of this research work are:

1. To measure the R-squared and Mean Squared Log Error for the proposed algorithm and compare the results with the existing system.
2. To compare the latency and power consumed by the proposed system when executed on a computer with an Intel 8265U processor and Raspberry Pi hardware.

2. Related Works

Many studies have addressed the normalized Kendall trend test coefficients with a spatial prediction scheme, kriging, and a stratified time series model for the prediction of time series pollutant data based on seasonal segregation and special patterns (Gan et al. [4]).

Feng et al. [5] proposed a dual-staged attention mechanism based on a conversion-gated Long Short-Term Memory (DACG-LSTM) network. In this work, an input gate and a forget gate are introduced with a hyperbolic tangent function to improve the ability to extract short-term mutation information. The RMSE for air quality is 0.28, MAE is 5.52, and MAPE is 0.19. From the results, it is inferred that the error is reduced compared to existing models, but it still persists in a considerable manner.

Munkhdalai et al. [6] developed an adaptive input selection mechanism for multivariate time series forecasting to improve prediction performance. This mechanism has two main stages. Inputs are dynamically selected by making the first-stage neural network learn about the generation of context-dependent importance weights. The target variable is then predicted by feeding the selected input into the second stage. Mean Absolute Percentage Error is recorded as 12.73 and 33.59 for NO2 and CO concentration, respectively.

Wang et al. [7] introduced a HI-VMD-SCA-ELM algorithm to improve the prediction performance of the Air Quality Index. The algorithm was tested in three different cities and produced a Mean Absolute Percentage Error and R-squared error of 6.0572 and 0.9836, respectively, for the city of Tianjin; 5.2574 and 0.9850, respectively, for the city of Shenyang; and 3.7078 and 0.9915, respectively, for the city of Shenzhen.

Zeng.Y et al. [8] proposed a Hybrid ESWT-NLSTM to analyze PM2.5 air quality. The hybrid model outperforms the benchmarking models by producing 11.61 and 0.990 as its MAPE and R-squared results, respectively. Ivanova.D et al. [9] proposed the development of an intelligent system that utilizes a machine learning model installed on the Raspberry Pi platform to predict various air pollutants based on temperature, humidity, and pressure.

Wang.Z et al. [10] investigated RLMD-ARIMA–RVMcom-MW to forecast PM2.5 and PM10 concentrations in four municipalities of China. The RLMD is used to decompose particulate matter time series into several production functions, which are then captured in both linear and nonlinear patterns in the subseries. The RMSE and MAPE values for PM2.5 are recorded as 2.9492 and 0.0816, respectively, and for PM10, they are recorded as 5.0223 and 0.0571, respectively.

Based on the survey, it is inferred that a Convolutional Neural Network-based model provides better accuracy when compared to various benchmarking models. Overall results obtained from the ARIMA model provide good accuracy in short-term prediction, and LSTM provides good results in long-term prediction. Nevertheless, previous articles still have shortcomings, such as ignoring the application of a totally versatile dataset in the proposed algorithm, realizing the proposed algorithm in a hardware module, escalating the model, and testing in a real-time environment, and so on [Wang.J et al. [7]]. So, in this work, the main objective is to propose a hybrid forecasting model to forecast pollutant concentrations by applying the algorithm to two different datasets. Primarily, the algorithm is applied with a dataset collected from NCR, India, and it is also tested using a dataset collected locally at the study site. In order to address real-time issues, our proposed algorithm is tested both with a computer with a system configuration of an Intel 8265U processor and also deployed in Raspberry Pi. The results are compared in terms of power consumption and latency.

3. System model

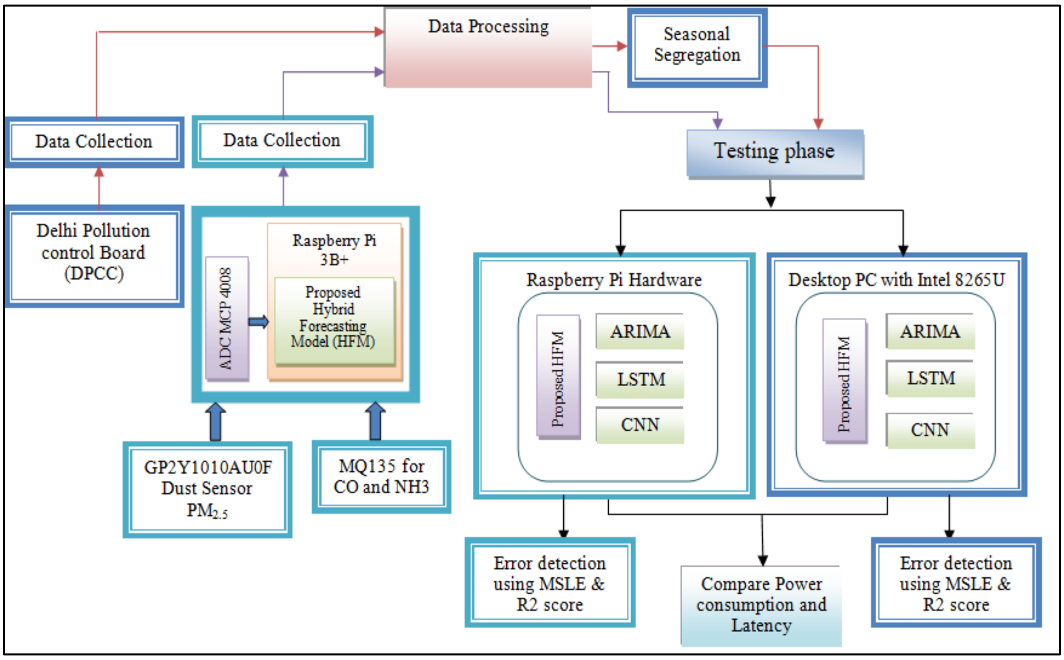

Figure 1. Proposed system model

The proposed system model is shown in Figure 1. The data collection phase plays a crucial role in this research. Pollutant concentrations are collected at two different data sites. Dataset 1, for the NCR, was collected from the Central Pollution Control Board over eight years. Dataset 2 comprises on-site pollutant concentrations collected with sensors integrated with hardware. The next step is to preprocess the data by replacing null values and detecting/correcting outliers.

In this work, Dataset 1 is seasonally segregated to improve air pollution forecasting by partitioning the highly non-stationary time series into quasi-stationary subsets with distinct meteorological characteristics. This reduces variations in statistical properties, stabilizes feature–target relationships, and enables models to capture season-specific patterns more effectively, thereby improving generalization accuracy.

The proposed Hybrid Forecasting Model (HFM) is evaluated against existing methods using Mean Squared Log Error (MSLE) and R^2. Furthermore, both the individual models and the HFM are implemented on an Intel 8265U-based system and a Raspberry Pi, with latency and power consumption analyzed to assess their suitability for real-time deployment.

3.1 Pseudocode- Proposed HFM

Procedure Model_1_ARIMA(x)

```
INPUT:
   Time series data x
OUTPUT:
   Optimized ARIMA model (model_opt)
aic_best ← ∞
model_opt ← NULL
FOR p ← 0 TO 3 DO
   FOR d ← 0 TO 2 DO
      FOR q ← 0 TO 3 DO
         model ← FIT_ARIMA(x, p, d, q,
                    allow_drift = TRUE,
                    allow_mean = TRUE)
         aic_curr ← COMPUTE_AIC(model)
         IF aic_curr < aic_best THEN
            aic_best ← aic_curr
            model_opt ← model
         END IF

      END FOR
   END FOR
END FOR

RETURN model_opt

END PROCEDURE
```

Procedure Model_2_LSTM(dataset)

```
INPUT:
   Raw dataset
OUTPUT:
   Trained LSTM model
dataset_norm ← NORMALIZE(dataset, range = [0,1])
train_set, test_set ← SPLIT(dataset_norm)
INITIALIZE LSTM network:
   input_units,
   lstm_units,
   output_units,
   optimizer
FOR epoch ← 1 TO EPOCHS DO
   FOR each batch in train_set DO
      TRAIN_LSTM(batch)
   END FOR
END FOR

RETURN trained_LSTM_model

END PROCEDURE
```

Procedure Model_3_Feature_Extraction(dataset, W)

```
INPUT:
   dataset,
   vectorization matrix W
OUTPUT:
   Feature matrix F
F ← EMPTY LIST
FOR each sample i in dataset DO
   fi ← EMPTY LIST
   FOR each element j in sample i DO
      vj ← VECTORIZE(j, W)
      APPEND vj TO fi
   END FOR
   APPEND fi TO F
END FOR
RETURN F
END PROCEDURE
```

Procedure Proposed_HFM(model1_pred, model2_pred, model3_pred, actual)

```
INPUT:
    Predictions from 3 models,
    actual values
OUTPUT:
    Ensemble predictions and evaluation metrics
N ← LENGTH(actual)
HFM_pred ← EMPTY LIST
FOR i ← 0 TO N-1 DO
    avg ← (model1_pred[i] + model2_pred[i] +
          model3_pred[i]) / 3
    APPEND avg TO HFM_pred
END FOR
// Evaluation
MSLE_HFM ← COMPUTE_MSLE(HFM_pred, actual)
R2_HFM ← COMPUTE_R2(HFM_pred, actual)
MSLE_M1, R2_M1 ← COMPUTE_METRICS(model1_pred, actual)
MSLE_M2, R2_M2 ← COMPUTE_METRICS(model2_pred, actual)
MSLE_M3, R2_M3 ← COMPUTE_METRICS(model3_pred, actual)
RETURN HFM_pred,
      (MSLE_HFM, R2_HFM),
      (MSLE_M1, R2_M1),
      (MSLE_M2, R2_M2),
      (MSLE_M3, R2_M3)

END PROCEDURE
```

3.2 Performance Metrics

Mean Squared Logarithmic Error (MSLE) is well-suited for air-pollution data because pollutant concentrations are non-negative, right-skewed, and vary across wide ranges. The log transformation stabilizes variance and reduces skewness, making the error distribution more uniform. MSLE emphasizes relative (percentage) errors, which are more meaningful than absolute errors in pollution levels. It is less sensitive to extreme spikes/outliers and penalizes underestimation more strongly. Thus, MSLE provides a robust, scale-aware evaluation, complemented by R-squared for overall fit. Therefore the proposed model's performance against existing models is evaluated using MSLE and R2 [11].

MSLE:

$$L(y,\hat{y}) = \frac{1}{N}\sum_{i=0}^{N}(\log(y_i+1)-\log(\hat{y}_i+1))^2 \tag{1}$$

R SQUARED:

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \hat{y})^2} \tag{2}$$

Where,

- N is the number of data samples
- y is the original data value
- $\hat{y}$ is the predicted data value.

4. Material and Methods

The proposed system model has five modules, namely, Data Set Description, Data Assimilation, Seasonal Segregation, proposed Hybrid Forecasting Module (HFM), and Hardware.

4.1 Dataset Description Module

In this research, two diversified datasets are analyzed and modeled using the proposed Hybrid Forecasting model. PM10, PM2.5, NO, and NO2 pollutant concentrations are collected for the National Capital Region (NCR), Delhi, from the Central Pollution Control Board (CPCB) [12], which is considered dataset_1. The detailed data analysis is given in Table 1. Train and test data are taken at a ratio of 80:20, respectively.

Table 1. Data analysis – dataset – 1

Season	2015	2016	2017	2018	2019	2020	2021	Total count
Cold Weather	2200	2200	2196	2197	2199	1840	2208	15,040
Hot Weather	2208	2199	2195	2197	2198	2199	2208	15,404
Advancing monsoon	2919	2920	2916	2917	2918	2920	2928	20,438
Retreating monsoon	1440	1416	1416	1416	1440	1416	1416	9960
Combined Count								60,842

A Raspberry Pi integrated with an MCP3008 ADC and pollutant sensors (MQ135 and GP2Y1010AU0F) was used to collect PM2.5, CO, and NH3 concentrations at the study site, forming dataset_2 [13]. A total of 17,289 samples were collected between September and October 2021 at Coimbatore Institute of Technology, Coimbatore, Tamil Nadu, India, at 5-minute intervals (283 samples/day). The urban study site, influenced by traffic, commercial, and residential emissions, is suitable for air quality monitoring and forecasting. The proposed

Hybrid Forecasting Model is evaluated using a government dataset and the on-site dataset_2 collected through the developed hardware setup Figures 2 and 3 show the NCR map and the on-site pollutant data collection location.

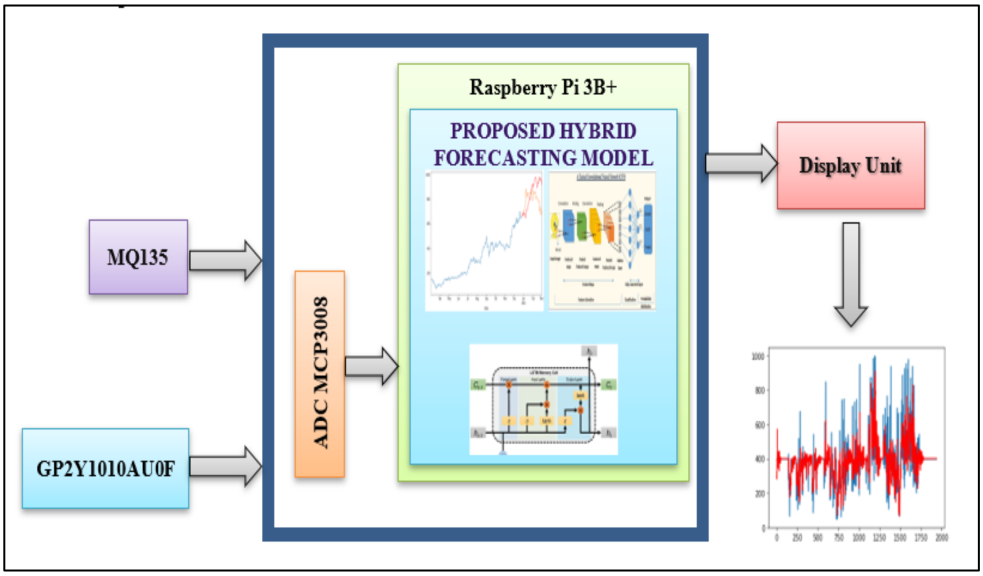

Figure 2. Hardware setup- Raspberry Pi with MCP3008

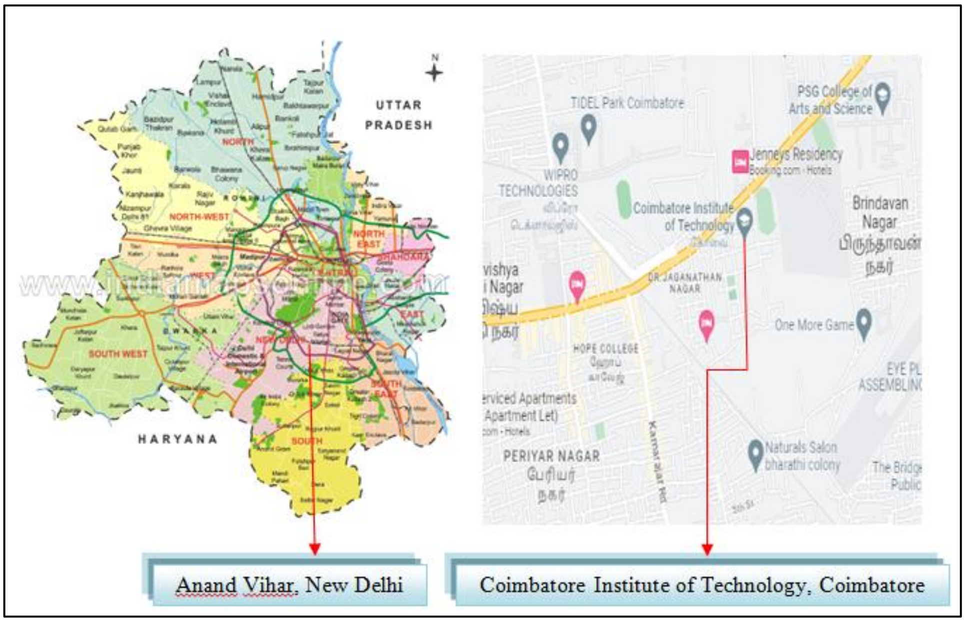

Figure 3. Study sites for data collection

4.2 Data Assimilation Modules

The collected data are processed using the proposed ensemble-based fusion model, where ARIMA, LSTM, and CNN are independently trained on the same temporally aligned dataset. Their predictions are combined through an optimally weighted aggregation scheme after normalization and time synchronization, enabling the model to capture both linear and nonlinear spatiotemporal patterns [14]. Since air pollution data may contain temporary irregularities, outliers are identified and removed to improve prediction accuracy [15].

$$\text{Inter-Quartile Range} \quad IQR = Q_3 - Q_1 \tag{3}$$

Lesser Outlier range= $Q_1 - 1.5 * IQR$ and

Greater Outlier range= $Q_3 + 1.5 * IQR$ (4)

Where,

- Q3 →3rd quartile of data
- Q1 →1st quartile of data

4.3 Seasonal Segregation module

In this research, dataset_1 has approximately 8 years of data collected from January 2015 to November 2022. Seasonal variations have a major impact on pollution levels, and to enhance forecasting performance, seasonal segregations are considered in this study. The Indian Meteorological Department, Government of India, located in Delhi, has divided India's climate into four seasons: Cold weather season (December to February), Hot weather season (March to May), Advancing Monsoon (June to September), and Retreating Monsoon Season (October to November) [16].

4.4 Proposed Hybrid Forecasting Module (HFM)

The proposed Hybrid Forecasting Module (HFM) is a combination of a statistical time series forecasting model (ARIMA), a Recurrent Neural Network-based LSTM, and an Artificial Neural Network-based CNN. Equation 5 shows the regression-type ARIMA model [10].

$$y'(t) = c + \phi_1 * y'(t-1) + ... + \phi_p * y'(t-p) + \theta_1 * \varepsilon(t-1) + ... + \theta_q * \varepsilon(t-q) + \varepsilon_t \quad (5)$$

Where,

y’(t) is the differenced series, p is the order of the lag, q is the order of the error lag, is the coefficient of the AR term and is the coefficient of the MA term, y(t-1), y(t-2) & y(t-p) are the regressed time series with its previous values and ε(t−1), ε(t−2) & ε(t−q) are the regressed time series with residuals of the past observations. The ARIMA parameters (p, d, q) are automatically determined using a stepwise search that ensures stationarity through differencing and selects optimal lag orders by minimizing information criteria like AIC.

AIC is comparable across multiple candidate models, which makes it practical during iterative model tuning.

The Recurrent Neural Network-based LSTM consists of three steps. Firstly, it checks whether the information from the previous timestamp is relevant or irrelevant. Secondly, it learns new information from the input to this cell. Finally, it updates the cell with the information from the current timestamp to the next timestamp.

The first step is to check whether the information is to be remembered or forgotten, as given in equation 6, which is evaluated using the weight matrix and timestamps of the input and hidden states [8].

$$f[t] = \sigma(X[t] * U_X + H[t-1] * W_H) \quad (6)$$

Where,

X[t] is the current timestamp's input, UX is the associated weight with the input, H[t-1] is the previous timestamp's hidden state and WH is the weight matrix related to the hidden state.

$$C[t-1] * f[t] = 0$$

if f[t]=0, indicates that the information is irrelevant

$$C[t-1] * f[t] = C[t-1]$$

if f[t]=1, indicates that the information is relevant

With this the input to the network is given in equation 7.

$$i[t] = \sigma(X[t] * U_i + H[t-1] * W_H) \tag{7}$$

Where, Ui is the input weight matrix

The second step is used learn new information from the input to this cell.

$$N[t] = \tanh(X[t] * U_c + H[t-1] * W_c)$$

Where,

tanh is the activation function in this case which ranges between -1 and +1. If N[t] is -1, the information is subtracted from the cell state and if N[t] is +1, information is added to the cell state at the current timestamp.

The third step is the output cell which updates the cell with the information from the current timestamp to the next timestamp. Equation 8 shows the output gate model and calculation of the current hidden state using tanh and O[t].

$$O[t] = \sigma(X[t] * U_O + H[t-1] * W_O) \text{ and } H[t-1] = O[t] * \tanh(C[t]) \tag{8}$$

The Artificial Neural Network based CNN can be greatly explained with the help of chain rule. Assume, E be the error function and the previous layer need to computed which is the partial of ε with respect to each neuron output $(\frac{\partial \varepsilon}{\partial y_{ij}^{l}})$ Yan. R et al [17].

The chain rule as shown in equation 9 adds the contribution of all expressions which has the presence of variables.

$$\frac{\partial \varepsilon}{\partial \omega_{ab}} = \sum_{i=0}^{N-m}\sum_{j=0}^{N-m} \frac{\partial \varepsilon}{\partial X_{ij}^{l}} \frac{\partial X_{ij}^{l}}{\partial \omega_{ab}} = \sum_{i=0}^{N-m}\sum_{j=0}^{N-m} \frac{\partial \varepsilon}{\partial X_{ij}^{l}} y_{(i+a)(j+b)}^{l-1} \tag{9}$$

Where, $y_{ij}^{l} = \sigma(x_{ij}^{l})$

Therefore,

$$\frac{\partial \varepsilon}{\partial X_{ij}^{l}} = \frac{\partial \varepsilon}{\partial y_{ij}^{l}} \frac{\partial}{\partial X_{ij}^{l}} (\sigma(X_{ij}^{l})) = \frac{\partial \varepsilon}{\partial y_{ij}^{l}} \sigma^{'} (X_{ij}^{l})$$

Equation 10 shows the model to propagate errors back to the previous layer.

$$\frac{\partial \varepsilon}{\partial y_{ij}^{l-1}} = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} \frac{\partial \varepsilon}{\partial X_{(i-a)(j-b)}^{l}} \frac{\partial X_{(i-a)(j-b)}^{l}}{\partial y_{ij}^{l-1}} = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} \frac{\partial \varepsilon}{\partial X_{(i-a)(j-b)}^{l}} \omega_{ab} \tag{10}$$

Where, $\frac{\partial X_{(i-a)(j-b)}^{l}}{\partial y_{ij}^{l-1}} = \omega_{ab}$

The proposed Hybrid Forecasting Model (HFM) is the mean of statistical time series forecasting models, namely the ARIMA model, the Recurrent Neural Network-based LSTM, and the Artificial Neural Network-based CNN. Equation 11 shows the proposed HFM model. The pollutant values are predicted based on the weight matrix, cell states, inputs, and error values. The ARIMA model captures linear trends, seasonality, and short-term autocorrelations in the data. The Long Short-Term Memory (LSTM) network models long-term temporal dependencies and nonlinear patterns that cannot be represented by linear models. The Convolutional Neural Network (CNN) extracts local temporal features and short-term fluctuations through convolutional filtering. By combining these models, the fusion framework leverages linear, nonlinear, and localized feature representations, resulting in improved predictive performance compared to any individual model.

$$y^{'}(t) = \frac{C + \phi_1 * y^{'}(t-1) + ... + \phi_p * y^{'}(t-p) + \theta_1 * \varepsilon(t-1) + ... + \theta_q * \varepsilon(t-q) + \varepsilon_t + \sigma(X[t] * U_O + H[t-1] * W_O) + \varepsilon^{'}(y_{ij}^{l}) \sigma^{'}(X_{ij}^{l})}{3} \tag{11}$$

Where,

Ɛ is the error, t denotes the current timestamp, (t-1) denotes the previous timestamp, p is the lag order and q is the error lag order, X is the input and y'(t) is the differenced series, is the coefficient of the AR term and is the coefficient of the MA term.

4.5 Hardware Module

The periodic recalibration or time-based calibration steps for the MQ135 and GP2Y1010AU0F sensor are discussed from equations 12 to 14 (Mani.G et al [18]).

$$CO(ppm) = (\frac{sensor_{data}}{1024}) * (\frac{R_L}{R_{CO}}) * (\frac{5.0}{V_{cc}}) * (\frac{1}{K}) \tag{12}$$

$$NH3(ppm) = (\frac{sensor_{data}}{1024}) * (\frac{R_L}{R_{NH3}}) * (\frac{5.0}{V_{cc}}) * (\frac{1}{K}) \tag{13}$$

$$PM_{2.5}(\frac{\mu g}{m^3}) = \frac{sensor_{data} - V_o}{K} \tag{14}$$

Where:

- $sensor_{data}$: the raw analog output from the sensor

- R_L: resistance of the load resistor (in ohms)
- R_{CO} and R_{NH3}: sensor resistance in clean air for CO and NH_3 respectively (in kilo-ohms)
- V_{cc}: supply voltage (in volts)
- K: sensitivity constant for the sensor (in ppm)
- V_o: sensor output voltage in clean air (usually around 0.6V)

5. Results and Discussion

In this work, two different datasets are analyzed, and results are compared with existing models and with the proposed hybrid forecasting model. The experimental process and the analysis of the forecasting results are discussed under two major sections. The first section deals with meteorological seasonal patterns-based forecasting results, and the second section deals with the voltage and temperature curve observed for the hardware module along with the forecasting results.

5.1 Meteorological Seasonal Patterns Forecasting Method Based on Multivariate Space-Time Hybrid Forecasting Model

To improve the accuracy of the forecasting results, outliers are eliminated from the collected dataset. In time-series air quality forecasting, outlier removal is applied to eliminate irregular, non-recurring events that can distort model learning. For example, sudden pollution spikes during festivals like Diwali or unusually low pollution levels during unexpected lockdowns do not represent normal environmental patterns. If such events are included, the model may learn misleading trends and produce inaccurate predictions under typical conditions. Therefore, these anomalies are treated as outliers to ensure the model captures consistent, real-world temporal behavior and improves overall prediction reliability. Figure 4 depicts the forecasting of the CO concentration test dataset. There is an uplift in the predicted output when compared to the input data, which is greatly eliminated by applying the outlier detection and correction algorithm. Thus, the input data and predicted output greatly coincide with each other, thereby portraying the importance of the outlier detection algorithm.

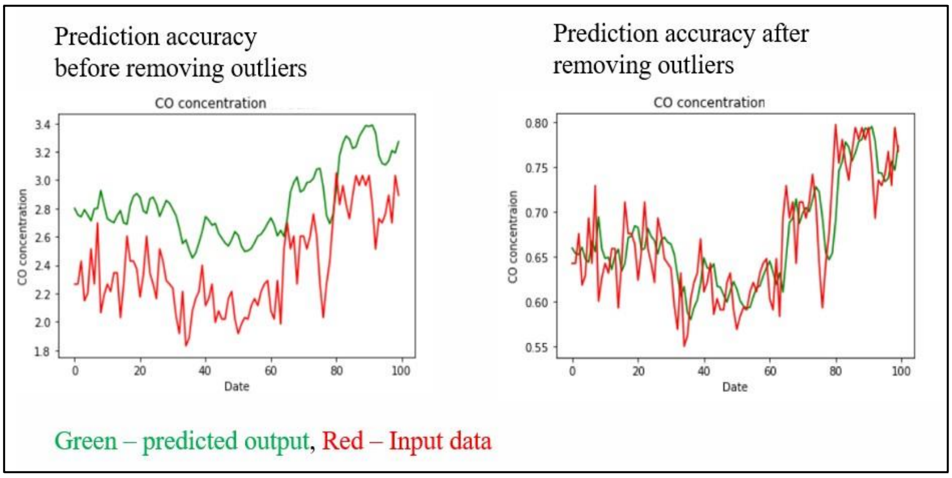

Figure 4. Comparison of accuracy with and without outliers

In this work, dataset_1 consists of PM10, PM2.5, NO, and NO2 pollutant concentrations from the National Capital Region (NCR), Delhi. The results of the proposed Hybrid Forecasting Model for all selected pollutants (PM10, PM2.2, NO2, and NO) are shown in Figure 5.

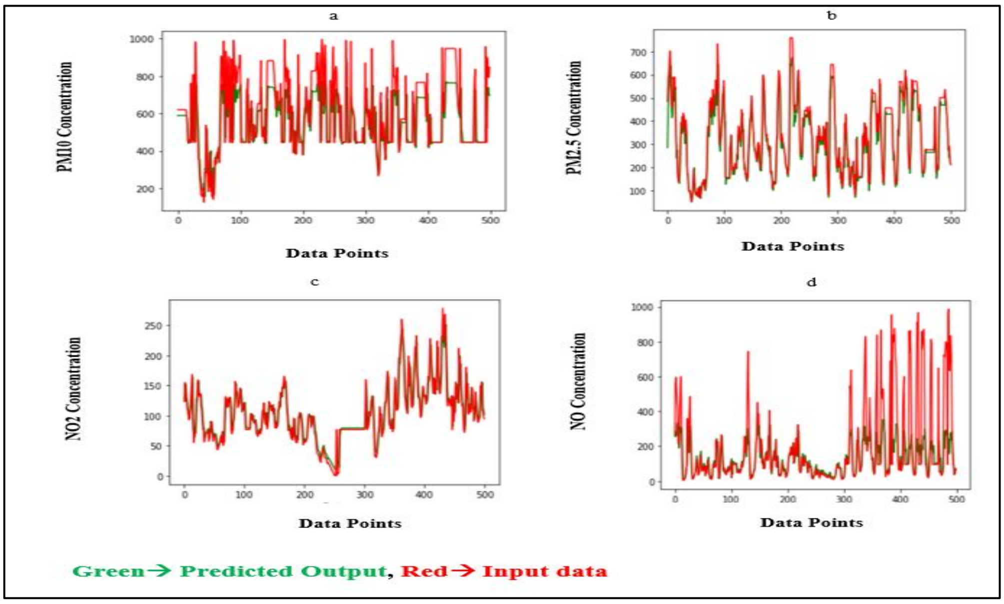

Figure 5. a) PM_{10} b) $PM_{2.5}$ c) NO_2 d) NO plots for the proposed HFM

The well-tamed dataset is segregated seasonally and applied to all the existing algorithms (CNN Yan.R et al. [17], RLMD-ARIMA–RVMcom-MW (Wang.Z et al. [10]), ESWT-NLSTM (Zeng.Y et al. [8])) and the proposed hybrid forecasting model. Figures [6] and [7] depict the output screenshots for the pollutant PM10 to enhance readability. The entire dataset underwent seasonal annulments. Figure 6 shows the proposed HFM plots for all four selected seasons. Figure 7 demonstrates the efficacy of the proposed HFM when compared to the existing systems.

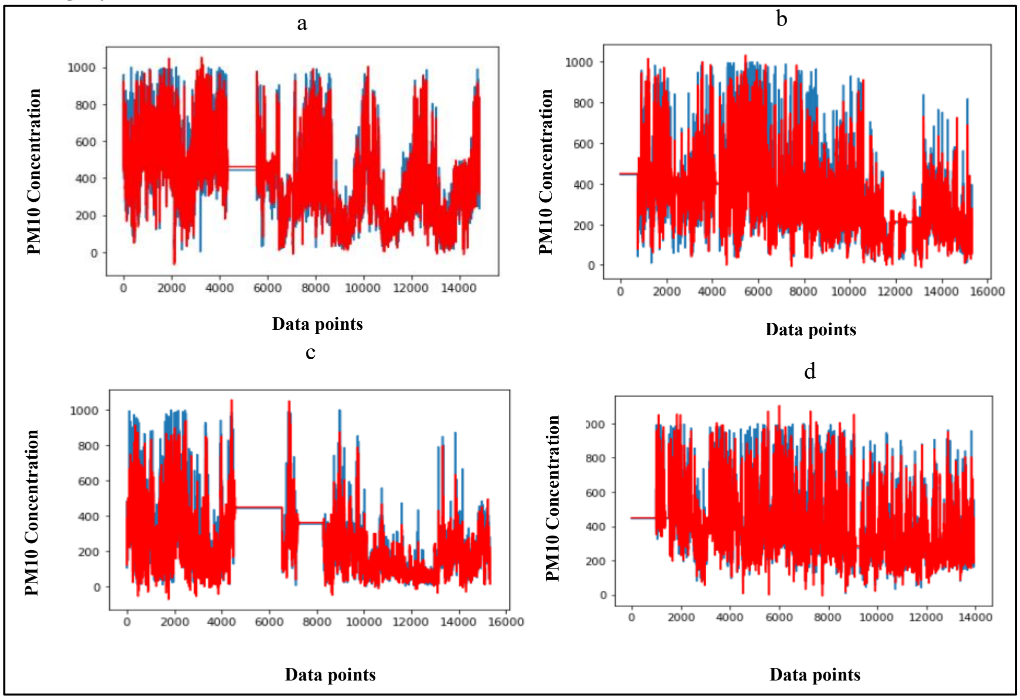

Figure 6. a) Autumn b) Spring c) Summer d) Winter plots for the proposed HFM

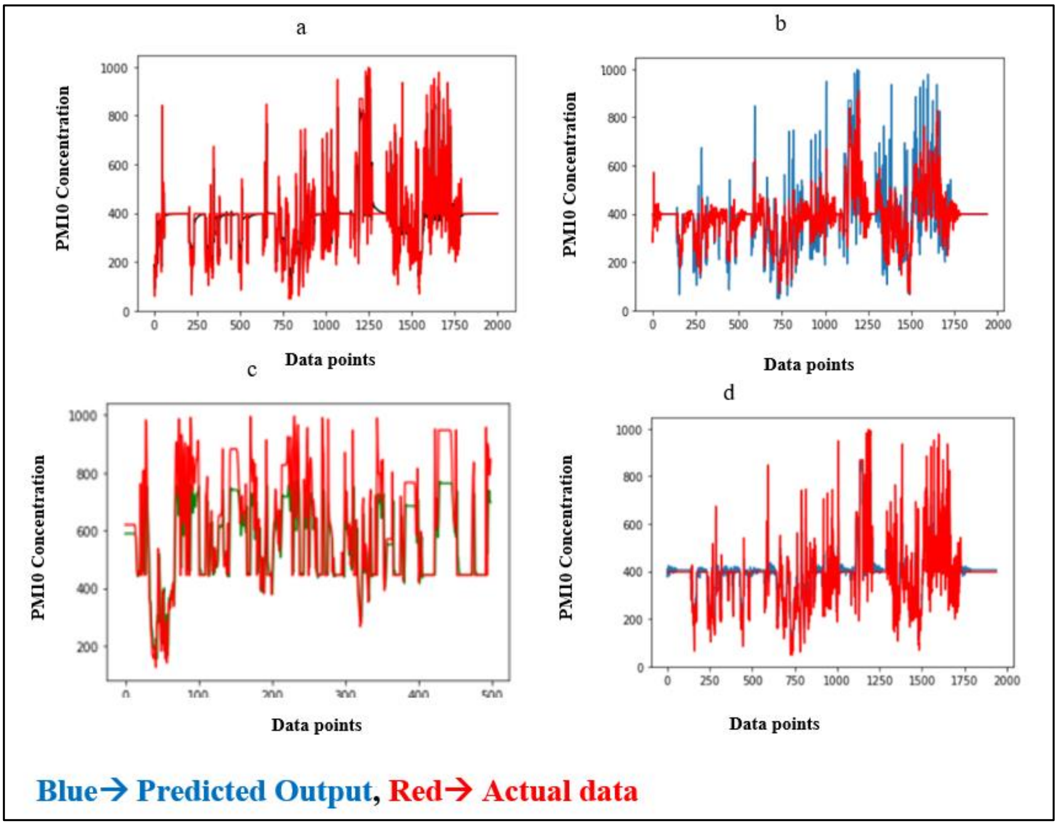

Figure 7. a) RLMD-ARIMA–RVM$_{com}$-MW b) CNN c) ESWT-NLSTM d) Proposed HFM plots for PM10

Table 2 shows the detailed results obtained by analyzing and evaluating the dataset using three existing systems and the proposed Hybrid Forecasting model. The type of dataset fed into the algorithms is mentioned in the first column. The "COMBINED" row indicates that 8 years of data have been combined, and the different pollutants are evaluated based on their MLSE values. Though CNN and LSTM perform well for {PM10:NO} and {PM2.5:NO2} respectively, the proposed HFM delivers minimal MLSE values for all pollutants. Similarly, the entire dataset underwent seasonal segregation, which was again analyzed and evaluated using existing and proposed HFM algorithms, and the results are tabulated in rows 3,4, 5, and 6. As the proposed HFM is modeled to adapt to changes with the seasonal patterns and trends and forecast future data points based on past timestamps, the proposed HFM shows the lowest MSLE values when compared to CNN (Yan.R et al [17]), RLMD-ARIMA–RVMcom-MW (Wang.Z et al [10]), and ESWT-NLSTM (Zeng.Y et al [8]) existing systems. This shows the fitness of the versatile dataset for the proposed HFM model.

Table 2. Comparison of MSLE values for all models and seasons

TYPE	MODEL	REFERENCE	PM10	PM2.5	NO	NO2
COMBINED	CNN	Yan.R et.al	0.002	0.043	0.242	0.037
	ARIMA	Wang.Z et al.	0.043	0.066	0.341	0.031
	LSTM	Zeng.Y et.al	0.014	0.041	0.486	0.024
	PROPOSED HFM		0.002	0.032	0.046	0.016
SUMMER	CNN	Yan.R et.al	0.239	0.489	0.298	0.068
	ARIMA	Wang.Z et al.	0.457	0.355	0.258	0.084
	LSTM	Zeng.Y et.al	0.278	0.282	0.323	0.052
	PROPOSED HFM		0.143	0.209	0.199	0.016
WINTER	CNN	Yan.R et.al	0.003	0.039	0.413	0.014
	ARIMA	Wang.Z et al.	0.049	0.076	0.393	0.035
	LSTM	Zeng.Y et.al	0.016	0.047	0.279	0.007

	PROPOSED HFM		0.003	0.038	0.196	0.005
AUTUMN	CNN	Yan.R et.al	0.021	0.046	0.488	0.016
	ARIMA	Wang.Z et al.	0.058	0.090	0.463	0.042
	LSTM	Zeng.Y et.al	0.019	0.056	0.329	0.011
	PROPOSED HFM		0.010	0.038	0.115	0.002
SPRING	CNN	Yan.R et.al	0.026	0.050	0.378	0.021
	ARIMA	Wang.Z et al.	0.074	0.096	0.328	0.039
	LSTM	Zeng.Y et.al	0.024	0.060	0.410	0.017
	PROPOSED HFM		0.022	0.032	0.309	0.015

Table 3 presents the R^2 scores of CNN (Yan R. et al. [17]), RLMD-ARIMA–RVMcom-MW (Wang Z. et al. [10]), ESWT-NLSTM (Zeng Y. et al. [8]), and the proposed Hybrid Forecasting Model (HFM) for the four seasons and the combined dataset. The proposed HFM achieves the best performance across all pollutants, demonstrating its robustness under varying environmental conditions. Table 4 summarizes the percentage improvement of HFM over the existing models. The relatively smaller improvements for certain pollutants are attributed to their low variability, strong dependence on external factors, and the effects of data preprocessing.

Table 3. Comparison of R2 score for all models and seasons

TYPE	MODEL	REFERENCE	PM10	PM2.5	NO	NO2
COMBINED	CNN	Yan.R et.al	0.990	0.934	0.925	0.890
	ARIMA	Wang.Z et al.	0.957	0.970	0.883	0.874
	LSTM	Zeng.Y et.al	0.942	0.979	0.346	0.942
	PROPOSED HFM		0.991	0.983	0.935	0.959
SUMMER	CNN	Yan.R et.al	0.895	0.815	0.846	0.922
	ARIMA	Wang.Z et al.	0.860	0.877	0.918	0.898
	LSTM	Zeng.Y et.al	0.852	0.878	0.334	0.891
	PROPOSED HFM		0.907	0.897	0.925	0.935
WINTER	CNN	Yan.R et.al	0.960	0.958	0.854	0.945
	ARIMA	Wang.Z et al.	0.925	0.918	0.973	0.931
	LSTM	Zeng.Y et.al	0.927	0.937	0.347	0.952
	PROPOSED HFM		0.975	0.964	0.981	0.962
AUTUMN	CNN	Yan.R et.al	0.931	0.948	0.854	0.949
	ARIMA	Wang.Z et al.	0.906	0.909	0.973	0.964
	LSTM	Zeng.Y et.al	0.956	0.927	0.474	0.971
	PROPOSED HFM		0.961	0.951	0.981	0.982
SPRING	CNN	Yan.R et.al	0.982	0.939	0.905	0.938
	ARIMA	Wang.Z et al.	0.915	0.965	0.982	0.904
	LSTM	Zeng.Y et.al	0.943	0.922	0.458	0.968
	PROPOSED HFM		0.995	0.974	0.997	0.973

Table 4. Proposed novel HFM- percentage of improvement in MSLE values and R2 score

TYPE	MSLE				R2 Score			
	PM10	PM2.5	NO	NO2	PM10	PM2.5	NO	NO2
COMBINED	0%	21.95%	81%	33.33%	0.10%	0.41%	1.08%	1.80%
SUMMER	40.17%	25.89%	22.87%	69.23%	1.34%	2.16%	0.76%	1.41%
WINTER	0%	2.56%	29.75%	28.57%	1.56%	0.63%	0.82%	1.05%
AUTUMN	47.37%	17.39%	65.05%	81.82%	0.52%	0.32%	0.82%	1.13%
SPRING	8.33%	36%	5.79%	11.76%	1.32%	0.93%	1.53%	0.52%

5.2 Hardware Module Results

In this research, dataset_2 comprises the on-site pollutant concentrations collected using sensors and a Raspberry Pi. The on-site CO, NH3, and PM2.5 pollutant data are gathered using an MQ135 gas sensor and a GP2Y1010AU0F dust sensor, respectively. These sensors are connected to the Raspberry Pi [19]. Since the Raspberry Pi cannot process analog data directly, an analog-to-digital converter IC, specifically the MCP3008 ADC, is integrated into the module. The hardware setup is illustrated in Figure 8.

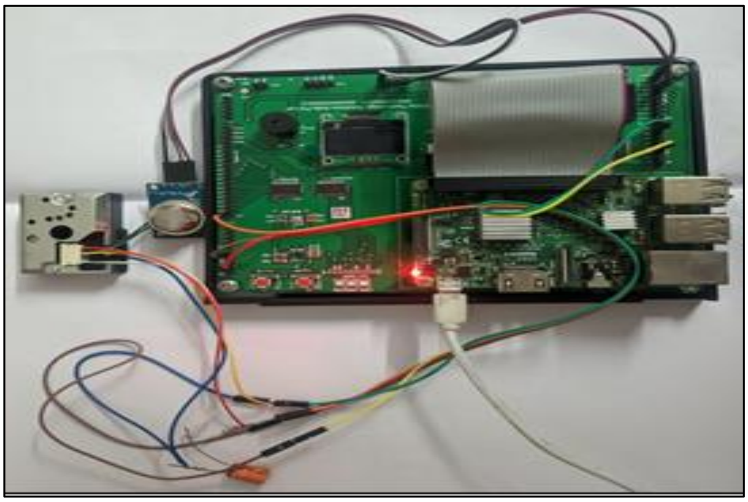

Figure 8. Hardware module

The calibration steps are discussed in equations 10 through 12. Data are collected from the calibrated sensors, which are downloaded as a CSV file. The proposed Hybrid Forecasting Model (HFM) runs on the OS, takes the CSV as its input, and forecasts the future values of pollutants. The forecasted results are shown in Figures 9a through 9c.

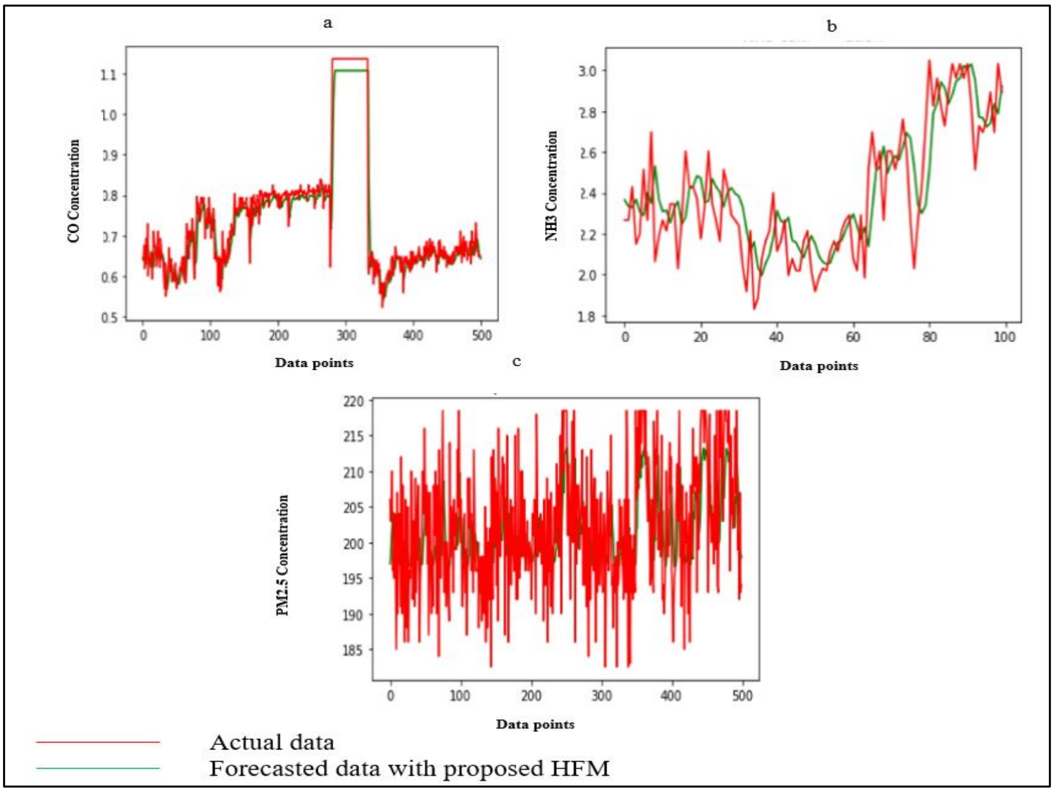

Figure 9. a) through c) Actual data vs proposed HFM forecasted data

From the experimental results, the ESWT-NLSTM [8] model outperforms the CNN [17] and RLMD-ARIMA–RVMcom-MW [10] models. Thus, a comparative study between actual data, ESWT-NLSTM [8], and the proposed HFM is done, which shows the perfectness of our proposed HFM model with the actual data. For simplicity, the PM2.5 concentration graph for the test data is shown in Figure 10.

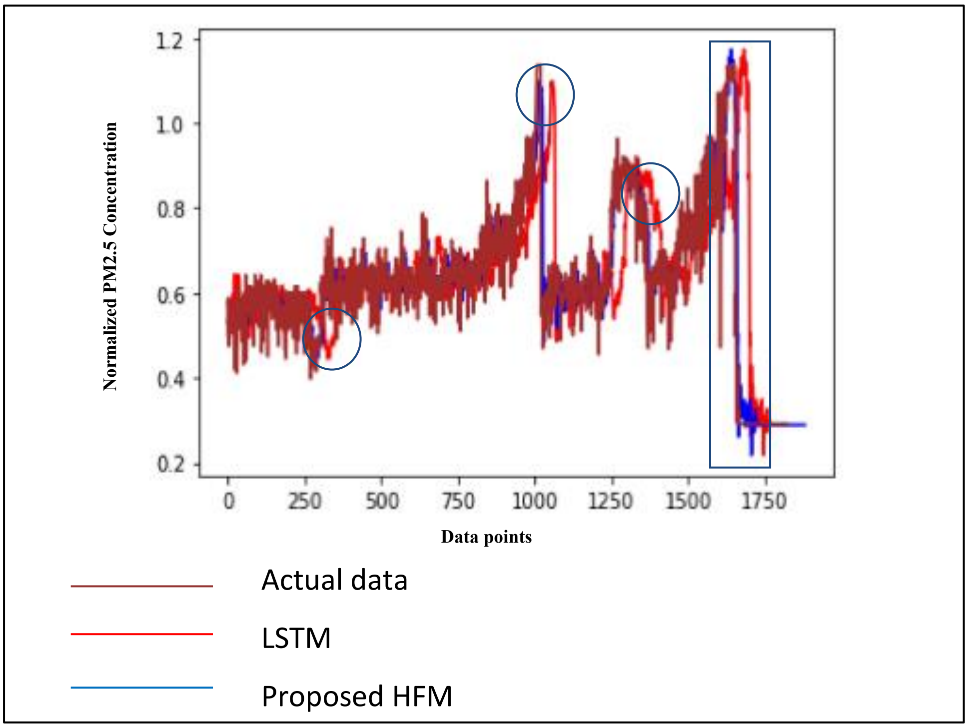

Figure 10. PM2.5 concentration forecasting

Table 5. Actual vs LSTM vs proposed HFM

Data Point	Actual Data	LSTM Prediction	Proposed HFM Prediction
~280	0.48	0.60	0.46
~1000	1.10	0.68	1.08
~1320	0.88	0.78	0.84
~1700	1.15	0.30	0.12

Table 5 shows the comparison of the Actual Data Pattern, LSTM (Bao.R et al [11]) and Proposed Hybrid Forecasting Model (HFM).

Table 6. Comparison of all the models based on performance metrics

METRICS	MODEL	PM2.5	CO	NH3
MLSE	CNN	0.059	0.098	0.147
	ARIMA	0.062	0.128	0.198
	LSTM	0.032	0.109	0.112
	PROPOSED HFM	0.030	0.072	0.110
R2	CNN	0.789	0.728	0.712
	ARIMA	0.745	0.798	0.658
	LSTM	0.897	0.872	0.765
	PROPOSED HFM	0.907	0.912	0.897

Table 6 shows the comparison results of all three benchmark models along with the proposed HFM. The results reveal that the proposed HFM outperforms all the benchmark models. In addition, the power consumption and latency are studied for a normal PC and Raspbian OS for the real-time implementation of the hardware module.

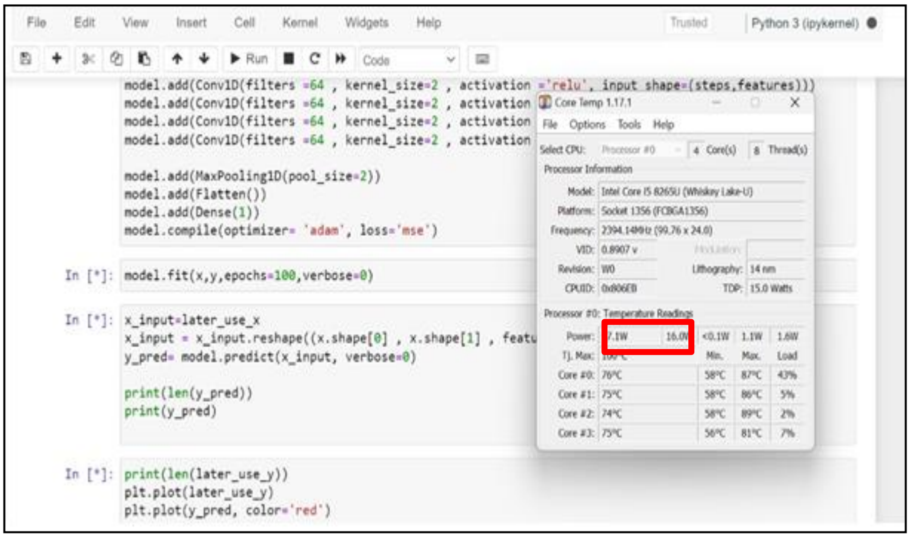

Figure 11. Power consumed by PC (Intel 8265 U)

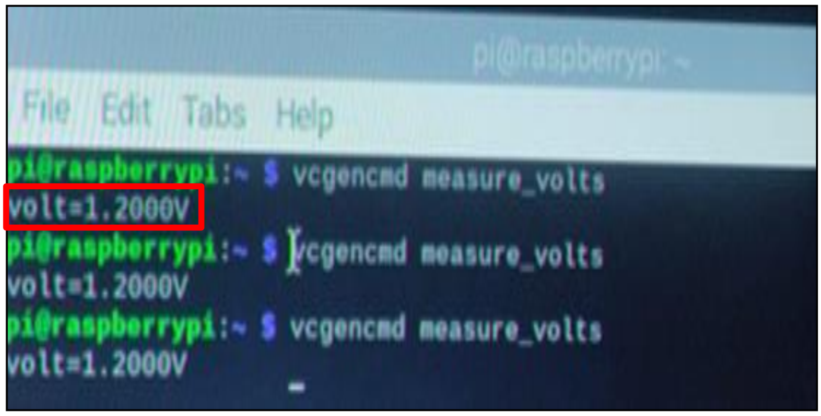

Figure 12. Voltage measured using Raspberry Pi

Figure 11 shows the power consumed by a PC using an Intel 8265U processor. Figure 12 shows the voltage measured using Raspberry Pi OS, whose current constant is 0.48 A.

Table 7. Comparison of Intel 8265U and Raspberry Pi

	Latency (in Sec)	Power Consumed (W)	Energy (J)
Intel 8265 U	8	17.1W	136.8
Raspberry Pi	60	0.576 W	34.56

Thus, in this research, the CPCB and sensor datasets are merged during the data processing stage through time-synchronized data fusion to show performance efficiency through an energy–latency trade-off. Hence, a comparison of Intel 8265U and Raspberry Pi, based on latency, power consumption, and energy, is shown in Table 7. From the table, it is inferred that the Intel 8265U provides low latency, which is suitable for real-time systems. On the other hand, its higher energy consumption makes it less efficient for continuous deployment. In contrast, Raspberry Pi shows higher latency, which results in slower inference, but consumes lower energy compared to Intel, making it ideal for Edge/IoT deployment.

Power Consumed Calculation:

$$= \frac{17.1 - 0.576}{17.1} X100$$

$$= \frac{16.524}{17.1} X100$$

$$= \sim 96.63\% \tag{15}$$

Latency Calculation:

$$= \frac{60-8}{60} X100$$

$$= \frac{52}{60} X100$$

$$= \sim 86.67\% \tag{16}$$

The Raspberry Pi consumes only approximately 3.37% of the power consumed by the Intel 8265U processor, making it highly power-efficient. On the other hand, the Intel 8265U processor provides approximately 86.67% faster performance with lower latency compared to the Raspberry Pi.

6. Conclusion

In this research work, the ESWT-NLSTM model outperforms CNN and RLMD-ARIMA–RVMcom-MW. These models are studied and compared with the proposed HFM model for two different datasets. The scope of the research is to demonstrate the HFM model's adaptability across two versatile and diverse datasets. Dataset_1 is fragmented season-wise to enhance prediction accuracy, while Dataset_2 focuses on the hardware implementation of the proposed algorithm using a Raspberry Pi 3 model. The highlights of the research are the versatility in pollutant selection, seasonal study, and hardware implementation of the proposed algorithm. The proposed algorithm is tested both on a computer with an Intel 8265U processor and on a Raspberry Pi. From the obtained results, it can be inferred that the Raspberry Pi consumes only approximately 3.37% of the power required by the Intel 8265U processor, highlighting its superior power efficiency. In contrast, the Intel 8265U processor demonstrates approximately 86.67% lower latency compared to the Raspberry Pi. The hardware model proposed in this research facilitates the real-time implementation of an air quality forecasting system in any zone of the city.

Acknowledgement

The data for this work was supported by Delhi Pollution Control Board (DPCB) from the website of Central Pollution Control Board (CPCB).

References

[1] Zhu, Suling, Jianan Sun, Yafei Liu, Mingming Lu, and Xingrong Liu. "The Air Quality Index Trend Forecasting Based on Improved Error Correction Model and Data Preprocessing for 17 Port Cities in China." Chemosphere 252 (2020): 126474.

[2] Wang, Deyun, Shuai Wei, Hongyuan Luo, Chenqiang Yue, and Olivier Grunder. "A Novel Hybrid Model for Air Quality Index Forecasting Based on Two-Phase

Decomposition Technique and Modified Extreme Learning Machine." Science of the Total Environment 580 (2017): 719-733.

[3] Wu, Qunli, and Huaxing Lin. "Daily Urban Air Quality Index Forecasting Based on Variational Mode Decomposition, Sample Entropy and LSTM Neural Network." Sustainable Cities and Society 50 (2019): 101657.

[4] Gan, Kai, Shaolong Sun, Shouyang Wang, and Yunjie Wei. "A Secondary-Decomposition-Ensemble Learning Paradigm for Forecasting PM2. 5 Concentration." Atmospheric Pollution Research 9, no. 6 (2018): 989-999.

[5] Feng, Shufang, and Yong Feng. "A Dual-Staged Attention Based Conversion-Gated Long Short Term Memory for Multivariable Time Series Prediction." IEEE Access 10 (2021): 368-379.

[6] Munkhdalai, Lkhagvadorj, Tsendsuren Munkhdalai, Kwang Ho Park, Tsatsral Amarbayasgalan, Erdenebileg Batbaatar, Hyun Woo Park, and Keun Ho Ryu. "An End-To-End Adaptive Input Selection with Dynamic Weights for Forecasting Multivariate Time Series." IEEE Access 7 (2019): 99099-99114.

[7] Wang, Jianzhou, Pei Du, Yan Hao, Xin Ma, Tong Niu, and Wendong Yang. "An Innovative Hybrid Model Based on Outlier Detection and Correction Algorithm and Heuristic Intelligent Optimization Algorithm for Daily Air Quality Index Forecasting." Journal of environmental management 255 (2020): 109855.

[8] Zeng, Yongkang, Jingjing Chen, Ning Jin, Xiaoping Jin, and Yang Du. "Air quality forecasting with hybrid LSTM and extended stationary wavelet transform." Building and Environment 213 (2022): 108822.

[9] Ivanova, Desislava, and Angel Elenkov. "Intelligent System for Air Quality Monitoring Assessment Using the Raspberry Pi Platform." In 2019 International Conference on Information Technologies (InfoTech), IEEE, 2019, 1-4.

[10] Wang, Zicheng, Huayou Chen, Jiaming Zhu, and Zhenni Ding. "Daily PM2. 5 and PM10 Forecasting Using Linear and Nonlinear Modeling Framework Based on Robust Local Mean Decomposition and Moving Window Ensemble Strategy." Applied Soft Computing 114 (2022): 108110.

[11] Bao, Ruiming, Yulong Zhou, and Wenrong Jiang. "FL-CNN-LSTM: Indoor Air Quality Prediction Using Fuzzy Logic and CNN-LSTM Model." In 2022 2nd International Conference on Electrical Engineering and Control Science (IC2ECS), IEEE, 2022, 986-989.

[12] Central Pollution Control Board. "Continuous Ambient Air Quality Monitoring Dashboard."https://airquality.cpcb.gov.in/ccr/#/caaqm-dashboard-all/caaqm-landing/data.

[13] Bhatti, Uzair Aslam, Yuhuan Yan, Mingquan Zhou, Sajid Ali, Aamir Hussain, Huo Qingsong, Zhaoyuan Yu, and Linwang Yuan. "Time Series Analysis and Forecasting of Air Pollution Particulate Matter (PM 2.5): An SARIMA And Factor Analysis Approach." Ieee Access 9 (2021): 41019-41031.

[14] Huang, Yuan, Yuxing Xiang, Ruixiao Zhao, and Zhe Cheng. "Air Quality Prediction Using Improved PSO-BP Neural Network." Ieee Access 8 (2020): 99346-99353.

[15] Shi, Weidi, and Anjun Song. "Air Quality Prediction Model Based on Spatio-Temporal Graph Convolution Neural Networks." In 2022 4th International Academic Exchange Conference on Science and Technology Innovation (IAECST), IEEE, 2022, 1015-1018.

[16] National Council of Educational Research and Training. India: Physical Environment (Chapter 4: Climate). New Delhi: NCERT, 2006.

[17] Yan, Rui, Jiaqiang Liao, Jie Yang, Wei Sun, Mingyue Nong, and Feipeng Li. "Multi-Hour and Multi-Site Air Quality Index Forecasting in Beijing Using CNN, LSTM, CNN-LSTM, And Spatiotemporal Clustering." Expert Systems with Applications 169 (2021): 114513.

[18] Mani, Geetha, Joshi Kumar Viswanadhapalli, and P. Sriramalakshmi. "AI Powered IoT Based Real-Time Air Pollution Monitoring and Forecasting." In Journal of Physics: Conference Series, vol. 2115, no. 1, IOP Publishing, 2021, 012016.

[19] Kiruthika, R., and A. Umamakeswari. "Low Cost Pollution Control and Air Quality Monitoring System Using Raspberry Pi for Internet of Things." In 2017 International conference on energy, communication, data analytics and soft computing (ICECDS), IEEE, 2017, 2319-2326.