ISSN: 2182-2069 / E-ISSN: 2182-2077

# An Approach towards Forecasting Time Series Air Pollution Data Using LSTM-based Auto-Encoders

Mohamed Shakir[1*], U. Kumaran[2], and Dr.N. Rakesh[3]

[1*]Research Scholar, Department of CSE, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Bengaluru, India. s_mohamed@blr.amrita.edu, https://orcid.org/0000-0001-6858-3220

[2]Assistant Professor, Senior Grade, Department of CSE, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Bengaluru, India. u_kumaran@blr.amrita.edu, https://orcid.org/0000-0002-0160-2703

[3]Associate Professor, Department of Information Science & Engineering, BMS Institute of Technology and Management, Bengaluru, Karnataka, India. n_rakesh@bmsit.in, https://orcid.org/0000-0001-8966-5831

Received: December 09, 2023; Revised: February 04, 2024; Accepted: March 05, 2024; Published: May 30, 2024

**Abstract**

Artificial Intelligence-based algorithm is used extensively for predicting the concentration of different pollutants under various conditions. Recently, Long-Short Term Memory (LSTM) and its variant is getting popular attention related to the prediction of Air Quality Index (AQI) across various polluted cities. The accuracy of the prediction is found to be depending on the processing step of input data. Here we present a study of combining both Random Forests (RF) based regression for data pre-processing step and multi-variate time series coupled with Multistep Multiwindow LSTM with auto encoder and decoder to predict the pollutant concentration in the urban city area of Bengaluru. In this approach, the RF algorithm is used for imputing the missing values of the input vector. We have implemented this technique from the data collected from the Karnataka State Pollution Control Board (KSPCB), for the city limits of Bengaluru with four years of data from 2019 to 2022, mainly focusing on the six regions where pollution is found to be maximum. We found that our Multistep Multivariate LSTM-based autoencoder gives better accuracy than the conventional LSTM data based on a single-step pipeline model with respect to Recall and F-Score values.

**Keywords:** LSTM, Auto-encoder, Time Series forecasting, Arithmetic Mean, Precision, Recall, F-Score.

## 1 Introduction

Air pollution is amongst the most grievous problems affecting the world today, and has a greater impact on human health and the economy. The pollution of the environment is caused when physical and biological components of land, water, and air are contaminated directly or indirectly which adversely affects the normal processes of the environment, resulting in biodiversity loss. Air pollution, water pollution, and land pollution together constitute environmental pollution. Pollution from noise, plastic,

*Journal of Internet Services and Information Security (JISIS)*, volume: 14, number: 2 (May), pp. 32-46.
DOI: *10.58346/JISIS.2024.I2.003*

*Corresponding author: Research Scholar, Department of CSE, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Bengaluru, India.

and light affect the biosphere as well (Brahmaiah et al., 2021; Chu & Karr, 2017; Niranjan & Rakesh, 2021; Pushpavalli et al., 2024). Air Quality Index (AQI) is decided depending on the concentrations of pollutants like Particulate Matter (PM10, PM2.5) and gases ($NO_2$, $O_3$, $CO_2$, $SO_2$, $NH_3$, Pb) suspended in the atmosphere as shown in Table 1.

Table 1: Air pollutants and their various ranges concerning Air Quality Index (AQI) categories [Courtesy: Central Pollution Control Board]

| AQI Category (Range) | PM10 24-hr | PM2.5 24-hr | $NO_2$ 24-hr | $O_3$ 8-hr | CO 8-hr (mg/meter cube) | $SO_2$ 24-hr | NH3 24-hr | Pb 24-hr |
|---|---|---|---|---|---|---|---|---|
| Good (0-50) | 0-50 | 0-30 | 0-40 | 0-50 | 0-1.0 | 0-40 | 0-200 | 0-0.5 |
| Satisfactory (51-100) | 51-100 | 31-60 | 41-80 | 51-100 | 1.1-2.0 | 41-80 | 201-400 | 0.6-1.0 |
| Moderate (101-200) | 101-250 | 61-90 | 81-180 | 101-168 | 2.1-10 | 81-380 | 401-800 | 1.1-2.0 |
| Poor (201-300) | 251-350 | 91-120 | 181-280 | 169-208 | 10.1-17 | 381-800 | 801-1200 | 2.1-3.0 |
| Very poor (301-400) | 351-430 | 121-250 | 281-400 | 209-748* | 17.1-34 | 801-1600 | 1201-1800 | 3.1-3.5 |
| Severe (401-500) | 430+ | 250+ | 400+ | 748+* | 34+ | 1600+ | 1800+ | 3.5+ |

To control and mitigate the ill effects of pollution, it is very much necessary to understand quantitatively the contents of different pollutants. A statistical way to analyze the prediction of air quality is by using time series analysis using Auto Regressive Integrated Moving Average (ARIMA) and Vector Autoregression (VAR) models respectively. It was observed that the model generated higher error values compared to the real measured values (Chu & Karr, 2017; Niranjan & Rakesh, 2020). Since the source and origin of pollutants are different, and the interaction with the atmospheric conditions are complex, currently, machine learning methodology is used widely for predicting air quality (Jurado et al., 2022). The multiplier-based enhancement in various machine learning algorithm has been proposed (Juma et al, 2023) research. One of the approaches currently used to predict air quality is the Long and Short-Term (LSTM) with various modifications (Sethi & Mittal, 2020; Camgozlu et al., 2023; Ram et al., 2024) The CNN network-based classification model is recently introduced in various fields. For instance, (Li & Hua, 2020; Shakir & Rakesh, 2018) envisioned a hybrid model of Convolutional Neural Network (CNN) and LSTM called as CNN-LSTM model for predicting the concentrations of PM2.5 values in the air measured on an hourly basis, from US Embassy in Beijing. These time series data were used to forecast PM2.5 using both the traditional LSTM and CNN-LSTM hybrid models. It turned out that the hybrid model took less time and had lower error rates for univariate and multivariate predictions (Setiawan & Setiawan, 2023; Kumar 2018; Kumaran, et al., 2021). Furthermore, (Niranjan & Rakesh 2021; Rakesh & Kumaran, 2021) developed a novel framework in accordance with Deep Learning (DL) method. This method is a combination of Multiple Nested CNN and conventional LSTM models. They have used nested LSTM (NLSTM), which combines additional LSTM units placed on each basal LSTM unit. A multi-modal network was proposed for forecasting mixed AQI data. Discrete stationary wavelet transform was used to convert raw data into multiple sub-signals at different frequencies. Different components of AQI were predicted and it was shown that their proposed model performed better than other methods in terms of operating time, lower error rates like Mean Absolute Error (MAE), Root Mean

Square Error (RMSE), Mean Absolute Percent Error (MAPE), and higher R-squared (R2) values (Jurado et al., 2022; Sharanyaa et al., 2022; Li & Wu, 2020; Arif et al., 2022).

In all the above models, the input time series data is normalized and the main multi-step LSTM is being implemented to predict the required parameters at various time steps. Though normalization makes the data simple to handle, it may trim the essential characteristics of parameters under consideration which may create the instabilities at later time steps. Using Random Forests (RF), a nonparametric statistical learning technique popular in many domains of application, including the analysis of microarrays, is one way to get around these instabilities. Díaz-Uriarte & Alvarez de Andrés (2006), ecology (Rakesh & Kumaran, 2021), pollution prediction (Yan et al., 2021; Kumaran et al., 2021). Dealing with missing data values is another significant difficulty in the handling of air quality data. There are several reasons why data sets contain missing information, including malfunctioning hardware, inadequate sample frequency, deteriorated equipment, and human mistake. (Norazian et al., 2008) To obtain a full set of data, one must choose whether to "impute" a phrase for replacing or reject the missing data. Inferential power may be compromised if important information is lost, hence it is usually not justified to ignore missing data. Impute the missing data is therefore the best course of action. However, unintended bias can also result from the systemic disparities between genuine and replaced data. Determining the best method for predicting missing values is so crucial. It is shown that the Missing at Random (MAR) based RF method gives fairly good accuracy for missing values, especially for air quality monitoring data sets (Alsaber et al., 2021). (Muralidharan, 2020), proposed the Multi Frequency resonator for remote based applications. Here, we have preprocessed and filled the missing values using MAR-based RF for multi time multivariate time series. It is to be noted thar the MAR-RF method goes through the interanion till it reaches the stopping criteria which prescribed based on the values of last imputed data. This preprocessed data is a component of the multistage LSTM which gives the multivariate, multi-time output from AQI.

In the next section 2, we will describe, the mathematical representation of time series including with RF-based algorithm for preprocessing the data. We will discuss the stopping criteria for the MAR-RF algorithm which we have followed to confirm the missing data is fairly accurate. In the same section we will describe our multistage LSTM in detail. In section 3 we will discuss in detail on the data which we have used to test our method. We will discuss the results of our prediction of AQI data in 4. Finally in section ?? we will summarize and conclude based on our results which we have generated through our approach and will discuss the future action planned of extending this research.

# 2 Theory and Methodology

**Random Forest (RF) Methodology for Pre-processing**

In this section we basically followed the methodology of (Prasad et al., 2006). Let $\mathbf{X} \in \mathbb{R}^{\times \mathbf{p}}$ and let Y be observed value of **X**, where there are missing values. We will discuss now the methodology by which RF technique will perform for imputing missing observations. First observed dataset is trained by random forest where missing values $\mathbf{X}_S \in i^S_{mis}$ .Therefore we end up with of datasets which are:

1. $\boldsymbol{Y^s_{obs}}$ is the set of all observed values of $\boldsymbol{X_s}$
2. $\boldsymbol{Y^s_{mis}}$ is the set of all observed missing of $\boldsymbol{X_s}$

which means $\boldsymbol{X^s_{obs}}$ is the dataset, where we will have the observation $i^S_{Obs}$= {1,…,n}\ $i^S_{mis} \in \boldsymbol{X_s}$.

We will start with the initial guess for the missing values in X as described in (Jin et al., 2021), depending on the data. Next, the missing values $\boldsymbol{Y^s_{mis}}$ are estimated by applying the trained random forest to $\boldsymbol{X^s_{mis}}$. This algorithm should be iterated until a stopping criterion is reached. The criteria for

stopping $\gamma_s$ is reached when, for both variable types, the difference between the current imputed data and the preceding one to prescribed precision. For a set of continuous variables N this is defined as:

$$\delta_{\mathbf{N}} = \frac{\Sigma_{j\in\mathbf{N}}\left(\mathbf{X}_{\text{new}}^{\text{imp}} - \mathbf{X}_{\text{old}}^{\text{imp}}\right)^2}{\Sigma_{j\in\mathbf{N}}\left(\mathbf{X}_{\text{new}}^{\text{imp}}\right)^2} \quad (1)$$

and that for the set of categorical variables **F** as:

$$\Delta_{\mathbf{F}} = \frac{\Sigma_{j\in\mathbf{F}}\,\Sigma_{i=1}^{n}\,\mathbf{I}_{\mathbf{X}_{\text{new}}^{\text{imp}} \neq \mathbf{X}_{\text{old}}^{\text{imp}}}}{N_A} \quad (2)$$

where $N_A$ is the number of missing values in the category $\mathbf{F}$.

In a nutshell, we follow the following steps:

Step 1: Set the stopping criterion $\gamma_s$.

Step 2: For missing values, make an assumption which is reasonable.

Step 3: Now we need to fit a random forest: such that $\mathbf{Y}_{\text{obs}}^{s} \sim \mathbf{X}_{\text{obs}}^{s}$ is satisfied.

Step 4: Predict $Y_{\text{miss}}^{s}$ using $X_{\text{miss}}^{s}$.

Step 5: Update the imputed values using the predicted $Y_{\text{miss}}^{s}$.

Step:6: Update $\gamma_s$ and $X^{imp}$.

The performance is assessed by using the normalized root mean squared error (NRMSE).

$$NRMSE = \sqrt{\frac{\left(\left(\mathbf{X}^{\text{true}} - \mathbf{X}^{\text{imp}}\right)^2\right)}{\text{var}(\mathbf{X}^{\text{true}})}} \quad (3)$$

All the missing AQI data is preprocessed by the above RF-based algorithm and it is represented by $\mathbf{X_t}$ This preprocessed data goes into our multistage LSTM algorithm, which we will discuss in the next sub-section.

**LSTM Methodology**

One of the favored approaches currently used in solving complex RNN problems is LSTM, which was originally proposed (Hochreiter & Schmidhuber, 1997) to mitigate vanishing gradient effects. In a nutshell, the recurrently connected subnets make up the LSTM architecture, which is known as a memory block. If we take the memory block of LSTM with one cell it has three gates: input, output, and forget. A schematic structure of a cell is shown in Figure 1. Let $\mathbf{X_t}$ represent processed data set at time t. In an LSTM network, the $\boldsymbol{U_f}$ in the forget gate equation (Stekhoven & Bühlmann, 2012) represents the weight matrix associated with the input connections to the forget gate. It indicates the amount of each element to retain. It plays a crucial role in mitigating the vanishing gradient problem and allowing LSTMs to capture long-term dependencies in sequences. The LSTM steps are as follows:

The Forget Gate $\mathbf{f_t}$ is expressed as:

$$\mathbf{f_t} = \sigma(\mathbf{W_f X_t} + \mathbf{U_f h_{t-1}} + \mathbf{b_f}) \quad (4)$$

The Input Gate $\mathbf{I_t}$ is expressed as:

$$\mathbf{I_t} = \sigma(\mathbf{W_i X_t} + \mathbf{U_i h_{t-1}} + \mathbf{b_i}) \quad (5)$$

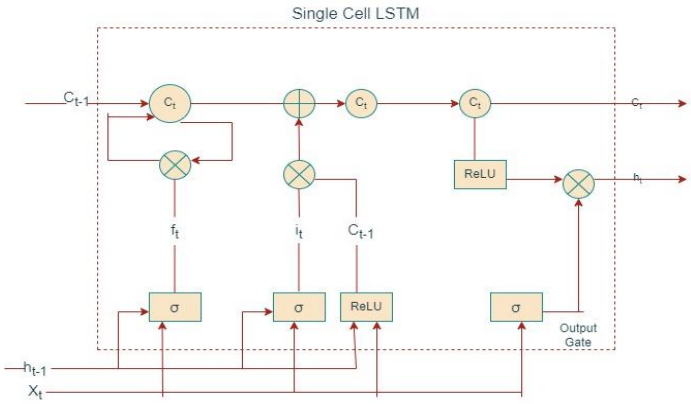



Figure 1: A schematic structure of a cell in LSTM

Output Gate $\mathbf{O_t}$ is expressed as:

$$\mathbf{O_t} = \sigma(\mathbf{W_o X_t} + \mathbf{U_o h_{t-1}} + \mathbf{b_i}) \quad (6)$$

Cell input activation vector:

$$\mathbf{C_t} = \mathbf{ReLU}(\mathbf{W}_c\mathbf{X_t} + \mathbf{U}_c\mathbf{h_{t-1}} + \mathbf{b_c}) \quad (7)$$

Cell state output:

$$\mathbf{C_t} = \mathbf{f_t} \otimes \mathbf{C_{t-1}} + \mathbf{i_t} \otimes \mathbf{C_t} \quad (8)$$

Hidden state Vector:

$$\mathbf{h_t} = \mathbf{O_t} \otimes \mathbf{ReLU}(\mathbf{C_t}) \quad (9)$$

where **b** is the bias and σ is the sigmoid function. Please note we have **ReLU** instead of tanh function which is used commonly in many LSTM models. Figure 2 shows the step of our algorithm. In the next section 3, we will elaborate on our data sets.

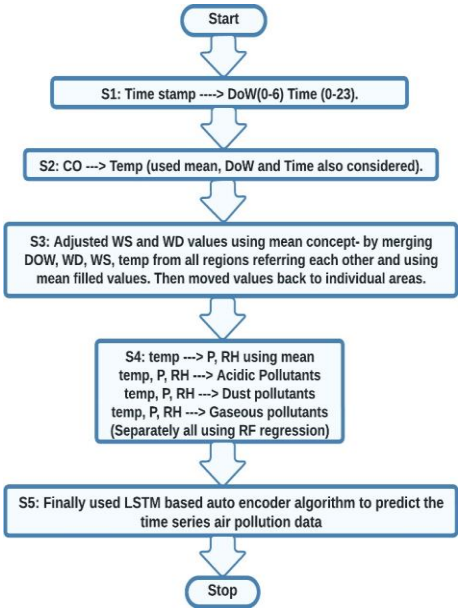



Figure 2: Flow Chart depicting the flow of tasks performed in making the prediction model

# 3 The Dataset

The dataset used in this research contains air quality and pollutant level data from Karnataka State Pollution Control Board, Bengaluru, collected in 6 regions of Bengaluru (Hebbal, Silk Board, NIMHANS, Kavika, Jaya Nagar, Majestic City Railway Station, and SG Halli Bangalore), over the last five years. The data collected from the first five regions contained 39000 records of hourly reports each. It consists of 14 fields. Table 2 gives the details of the various air pollutants (features) under study. Imputation of missing values was done at three stages based on the type of pollutants viz., gaseous, chemical, dust since the type of values for each feature varies integer or float, etc., the forecasting was carried out using LSTM-based Autoencoder. Our model configuration is shown in the Figure 1 (Shakir & Rakesh, 2018).

Air quality data was gathered from 7 locations in Bengaluru City. Table 1 lists the pollutant level and the environmental parameter information collected during the process. Information collected from the Hebbal, Jaya Nagar, Kavika, NIMHANS, and Silk Board areas consists of hourly data for the last five years, while that gathered from City Railway Station and S.G Halli areas consists of the daily report for the last five years. The Day of Week (DOW) component derives the 'day of the week', 'hour of the day', 'date', and 'time' details from the time stamp field and adds them to the dataset.

Table 2: Various features under study that are classified into three categories - gaseous, dust, and chemical pollutants

| Sl. No. | Feature Name | Type | Unit |
|---|---|---|---|
| 1 | Gaseous Pollutants | CO | mgm/m3 |
| 2 | | $O_3$ | µgm/m3 |
| 3 | | NO | µgm/m3 |
| 4 | | $NO_2$ | µgm/m3 |
| 5 | | NOx | µgm/m3 |
| 6 | | $NH_3$ | µgm/m3 |
| 7 | | $SO_2$ | µgm/m3 |
| 8 | Dust Pollutants | PM2.5 | µgm/m3 |
| 9 | | PM 10 | µgm/m3 |
| 10 | Acidic Pollutants | Benzene | µgm/m3 |
| 11 | | Toluene | µgm/m3 |
| 12 | | m,p-Xylene | µgm/m3 |
| 13 | | o-Xylene | µgm/m3 |
| 14 | | Ethyl Benzene | µgm/m3 |

The enhanced dataset from each area is then passed to the next stage, where missing temperature values are filled using values from the CO field, using the arithmetic mean. The next step grouped the DOW, hour, time stamp, ambient temperature, wind speed, and wind direction fields from Hebbal, Jaya Nagar, Kavika, NIMHANS, and Silk Board areas to address the missing values in wind speed and wind direction fields. It used the arithmetic mean to fill in the missing values. The updated values were moved back to the original datasets. The next stage addressed the relative humidity, and the barometric pressure in the individual datasets using the arithmetic mean. The next stage used the Random Forest regression to address the missing values in the pollutant's fields. The main algorithm involves the use of an LSTM-based Auto encoder time series algorithm to forecast the values of pollutants. The algorithm used

a input sequences of 10 time steps of input variables as input windows, predicted a sequence of 5 time steps of output(10X5). This research evaluated two variations of autoencoders, with single LSTM and dual LSTM layers in the encoder and decoder stages. The proposed research work evaluated two versions of LSTM based Auto encoders the first having just one concealed layer in the encoder and decoder, and the second with double hidden layers. The model was trained using different parameters as shown in Table 3.

Table 3: Listing the various parameters and their corresponding values used during model creation

| Sl. No. | Parameter | Value |
|---|---|---|
| 1 | Number of neurons | 100 |
| 2 | Number of epochs | 50 |
| 3 | batch size | 32 |
| 4 | Window size | 10X5 |

# 4 Result and Discussion

Based on the discussion above in Section 3, we have simulated the following 5 different configurations:

Configuration 1 LSTM-based Autoencoders – Single hidden layer
Configuration 2 LSTM-based Autoencoders – Single hidden layer
Configuration 3 LSTM-based Autoencoders – Double hidden layers
Configuration 4 LSTM-based Autoencoders – Single hidden layer (only 2 features AT & CO)
Configuration 5 LSTM-based Autoencoders – Single hidden layer

To make our report concise, we are presenting the result only for CO-polluted concentration. It is implied that we can extend this methodology to predict other pollutants also without any difficulty.

The efficiency metrics of our simulated results produced are evaluated with respect to three parameters namely, precision, recall, and F1-score values.

1. **Precision:** Is a classification model's capacity to recognize only pertinent data points. The number of true positives divided by the total number of true positives + the number of false positives.
2. **Recall:** Is a metric that represents the proportion of positive cases, out of all the positive cases in the data, that the classifier correctly predicted. Another name for it is sensitivity at times. It is the proportion of accurate predictions to all data sets that fall into that class.
3. **F1–score:** Values is The harmonic mean of recall and precision is called F1-score. It provides a balanced score for precision and recall. The F1 will be high only when both precision and recall are high.

Table 4 sums up the results of our study for four different configurations including **loss function**. To optimize the present state, we need to estimate the error of the model which is by **loss function**

Table 4: Accuracy Precision, Recall, F-score and Label Value for various con- figuration

| Configuration Setup No. | Number of neurons | Window Size | Precision obtained | Recall value | F-Score value | Loss value |
|---|---|---|---|---|---|---|
| 1 | 100 | 10X5 | 1.0 | 0.97 | 0.98 | 5.92 |
| 2 | 150 | 10X5 | 1.0 | 0.98 | 0.99 | 4.17 |
| 3 | 100 | 10X5 | 1.0 | 0.95 | 0.97 | 6.99 |
| 4 | 100 | 10X5 | 1.0 | 0.79 | 0.88 | 0.27 |
| 5 | 100 | 5X3 | 1.0 | 0.95 | 0.97 | 4.15 |

From the above analysis, it is inferred that configuration set up 4 which having a single-layered LSTM-based Autoencoder with 100 neurons and window size of 10X5 and with just 2 features (CO and AT) has lower Recall and F1-Score values. Unlike other setups, the loss incurred is less since the dimensions of the dataset were reduced to just two features. The performance configuration 2, with single layered LSTM-based Auto Encoder having 150 neurons; a window size of 10X5 performed better compared to other models with the highest Recall and F-Score values and comparatively small loss value. The model configurations discussed above used the Adam optimizer, Huber loss function, and ReLu activation in the LSTM layers. Table 4, 5, and 6 summarizes the results of our simulation. It is seen that the LSTM-based Auto Encoder model with various configuration setups has predicted the values near the actual outputs.

We have also found that the loss value reduces and remains constant after 30 epochs for all the proposed models under various configurations as shown in Figures 3, and 4.

Table 5: Actual values as per the dataset

| Type 10X5 window size | Date/ Time (Index) | Atmospheric Temperature | Relative Humidity | Barometric Pressure | CO |
|---|---|---|---|---|---|
| Actual Input | 2023-4-17 09:45:00 | 25.1 | 71 | 47 | 0.6 |
| | 2023-4-17 10:00:00 | 27.9 | 71.2 | 47 | 0.55 |
| | 2023-4-17 10:15:00 | 27.9 | 71 | 43 | 0.59 |
| | 2023-4-17 10:30:00 | 27.9 | 71 | 41 | 0.51 |
| | 2023-4-17 10:45:00 | 27.9 | 71 | 38 | 0.51 |
| | 2023-4-17 11:00:00 | 32 | 71 | 36 | 0.49 |
| | 2023-4-17 11:15:00 | 32 | 71 | 34 | 0.39 |
| | 2023-4-17 11:30:00 | 32 | 71 | 30 | 0.33 |
| | 2023-4-17 11:45:00 | 32 | 71 | 28 | 0.42 |
| | 2023-4-17 12:00:00 | 34.9 | 71 | 26 | 0.51 |
| Actual Output | 2023-4-17 12:15:00 | 34.9 | 71 | 26 | 0.47 |
| | 2023-4-17 12:30:00 | 34.9 | 71 | 27 | 0.36 |
| | 2023-4-17 12:45:00 | 34.9 | 71 | 28 | 0.53 |
| | 2023-4-17 01:00:00 | 36.3 | 71 | 28 | 0.58 |
| | 2023-4-17 01:15:00 | 36.3 | 71 | 28 | 0.6 |

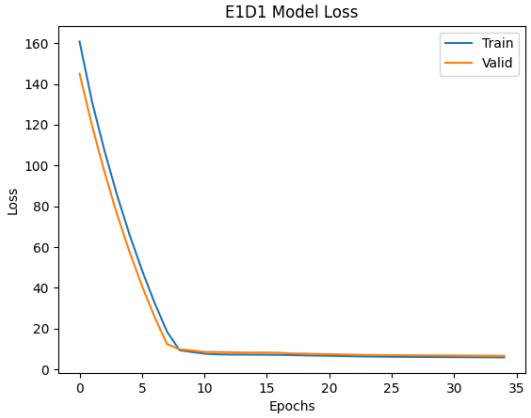
Table 6 : Predicted values using various configuration models

| Type 10x5 Window Size | Date/ Time (Index) | Atmospheric Temperature | Relative Humidity | Barometric Pressure | CO |
|---|---|---|---|---|---|
| Predicted Output for Configuration 1 Multivariate e1d1 100 neurons algorithm | 2023-4-17 12:15:00 | 35.947 | 68.773 | 26.78 | 0.4841 |
| | 2023-4-17 12:30:00 | 33.853 | 73.02 | 26.19 | 0.3708 |
| | 2023-4-17 12:45:00 | 33.853 | 73.03 | 28.84 | 0.5141 |
| | 2023-4-17 13:00:00 | 35.211 | 68.8 | 27.16 | 0.5974 |
| | 2023-4-17 13:15:00 | 37.389 | 73.24 | 28.84 | 0.618 |
| Predicted Output for Configuration 2 Multivariate e1d1 150 neurons algorithm | 2023-4-17 12:15:00 | 35.598 | 69.5 | 26.52 | 0.4794 |
| | 2023-4-17 12:30:00 | 35.598 | 72.3 | 26.46 | 0.3672 |
| | 2023-4-17 12:45:00 | 34.202 | 72.3 | 28.56 | 0.5194 |
| | 2023-4-17 13:00:00 | 37.026 | 69.5 | 27.44 | 0.5916 |
| | 2023-4-17 13:15:00 | 35.574 | 69.3 | 28.56 | 0.588 |
| Predicted Output for Configuration 3 Multivariate e2d2 100 neurons algorithm | 2023-4-17 12:15:00 | 33.504 | 73.7 | 27.04 | 0.4512 |
| | 2023-4-17 12:30:00 | 36.296 | 68.54 | 28.08 | 0.3744 |
| | 2023-4-17 12:45:00 | 36.296 | 68.1 | 26.88 | 0.5088 |
| | 2023-4-17 13:00:00 | 34.848 | 73.7 | 26.88 | 0.6032 |
| | 2023-4-17 13:15:00 | 37.752 | 73.6 | 29.12 | 0.576 |
| Predicted Output for Configuration 4 Multivariate e1d1 100 neurons algorithm only 2 features - AT and CO | 2023-4-17 12:15:00 | 39.0531 | - | - | 0.41407 |
| | 2023-4-17 12:30:00 | 39.0531 | - | - | 0.40284 |
| | 2023-4-17 12:45:00 | 30.7469 | - | - | 0.59307 |
| | 2023-4-17 13:00:00 | 40.6197 | - | - | 0.51098 |
| | 2023-4-17 13:15:00 | 31.9803 | - | - | 0.5286 |
| Predicted Output for Configuration-5 Multivariate e1d1-5X3_ws_algorithm | 2023-4-17 12:15:00 | 36.296 | 73.84 | 27.04 | 0.4888 |
| | 2023-4-17 12:30:00 | 35.947 | 73.13 | 27.81 | 0.3708 |
| | 2023-4-17 12:45:00 | 33.504 | 68.16 | 26.88 | 0.5088 |
| | 2023-4-17 13:00:00 | 35.211 | 68.87 | 27.16 | 0.5626 |
| | 2023-4-17 13:15:00 | 37.026 | 72.42 | 28.56 | 0.612 |

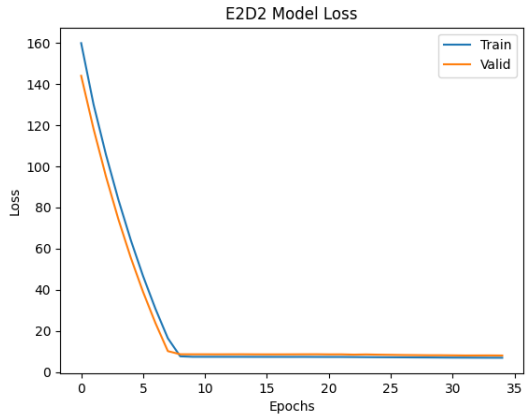
Table 5 presents the actual input and output values obtained from observations and helps in predicting values generated by the LSTM-based Auto-Encoder model with different configuration setups. This table includes specific parameters related to atmospheric conditions and pollutant concentrations, such as atmospheric temperature, relative humidity, barometric pressure, and CO concentration. The structured format likely includes columns representing the date and time of observations, atmospheric temperature, relative humidity, barometric pressure, and CO concentration, with the actual values recorded for each parameter. By comparing the actual values with the predicted values, we can assess the model's performance in accurately forecasting pollutant levels based on the input data. This comparison is valuable for evaluating the model's accuracy, reliability, and effectiveness in predicting environmental parameters essential for air quality monitoring and pollution control efforts. Therefore, Table 5 serves as a critical tool for validating the model's predictive capabilities and enhancing its applicability in environmental research and decision-making processes.

Table 6 presents the efficiency metrics of the simulated results produced by the LSTM-based Auto-Encoder models with different configuration setups. Specifically, the table showcases the precision, recall, and F1-score values for each model configuration, serving as indicators of the model's accuracy in predicting pollutant levels. The results demonstrate that the LSTM-based Auto-Encoder model, across various configuration setups, has successfully predicted values that closely align with the actual outputs.

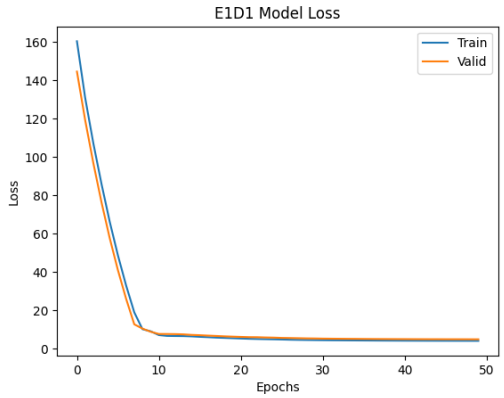
In summary, these tables offer a comprehensive overview of the simulation results, highlighting the efficacy of the LSTM-based Auto-Encoder models in forecasting pollutant levels in urban areas.



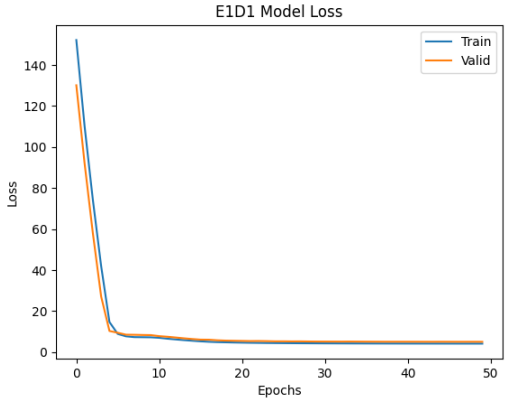

(a) LSTM Auto-Encoder with single layer




(b) LSTM-based Auto-Encoder with double layers




(c) LSTM-based Auto-Encoder with a single layer having 100 neurons in the hidden layer




(d) LSTM-based Auto-Encoder with a single layer having 100 neurons in the hidden layer having only two features

Figure 3: Demonstration of the loss values vs. the number of epochs for configurations 1-4

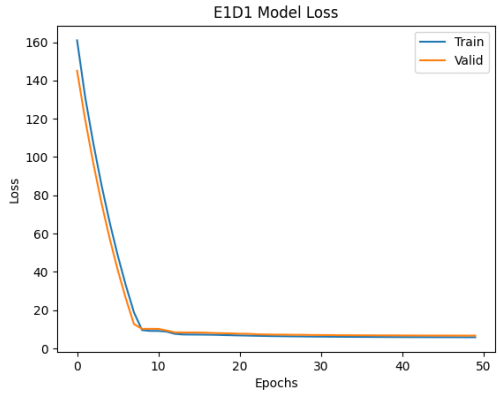



Figure 4: LSTM-based Auto-Encoder with a single layer having 100 neurons in the hidden layer for configuration 5

Figure 3 above likely depicts the correlation between the loss values and the number of epochs for configurations 1 to 4 of the LSTM-based Auto-Encoder algorithm. The horizontal axis of the graph represents the progression of training epochs, which are iterations over the dataset during the model training phase. On the vertical axis, the loss values are plotted, indicating the model's performance at each epoch. Lower loss values signify better model performance and accuracy.

Each configuration (1 to 4) represented by distinct lines or curves on the graph, illustrates how the loss values evolve with increasing epochs. This visualization aids in assessing the training dynamics of the models. Ideally, a downward trend in loss values across epochs suggests that the model is learning and enhancing its predictive capabilities.

By analysing Figure 3, we can compare the training patterns of different configurations and identify which configuration – 4 setup leads to the most significant reduction in loss values throughout the training process, since it has only two features.

Figure 4 illustrates the performance of the LSTM-based Auto-Encoder algorithm with a single hidden layer containing 100 neurons and window size 5X3 for configuration 5. The graph shows the relationship between the loss values and the number of epochs during the training phase.

By analyzing Figure 4, we can evaluate the effectiveness of the LSTM-based Auto-Encoder algorithm with a single hidden layer containing 100 neurons 5X3 window size for configuration 5 in reducing loss values over the training epochs and it is almost equal to configuration 2 set with single layer and 150 neurons as depicted in table-4.

From fig. 3 and fig. 4, it is clear that the loss value reduces and remains constant after 30 epochs for all the proposed models under various configurations.

## 5 Summary and Conclusion

This research work compared the performance of an LSTM-based Auto encoder algorithm under different model configurations to forecast the pollutant levels based on time series data. The data was given by the Karnataka State Pollution Control Board (KSPCB), Bengaluru. Imputation of missing values was carried out with a novel methodology by considering the practical correlation between pollutants and atmospheric features. Arithmetic mean and Random Forest regression were used to fill in the missing values. Moreover, imputation of missing values was done at three stages based on the type of pollutants viz., gaseous, chemical, dust since the type of values for each feature varies integer or float, etc., the forecasting was carried out using LSTM-based Autoencoder. The results of various models built using LSTM-based Auto Encoders have been analyzed. Future research will evaluate the impact of pollution in one region on the neighboring region.

## Acknowledgment

We express our gratitude to the Karnataka State Pollution Control Board (KSPCB), Bengaluru, for providing the necessary dataset for this study.

## References

[1] Alsaber, A.R., Pan, J., & Al-Hurban, A. (2021). Handling complex missing data using random forest approach for an air quality monitoring dataset: a case study of Kuwait environmental data (2012 to 2018). *International Journal of Environmental Research and Public Health*, *18*(3), 1333. https://doi.org/10.3390/ijerph18031333

[2] Arif, K.D., Volkan, R., Mert, N., Buse, P., & Cuneyt, G. (2022). Multi-Channel Subset Iteration with Minimal Loss in Available Capacity (MC-SIMLAC) Algorithm for Joint Forecasting-Scheduling in the Internet of Things. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, *13*(2), 68-95.

[3] Brahmaiah, B., Vivek, G.V., Gopal, B.S.V., Sudheer, B., & Prem, D. (2021). Monitoring And Alerting System based on Air, Water and Garbage Levels Using Esp8266. *International Journal of Communication and Computer Technologies (IJCCTS), 9*(2), 31-36.

[4] Camgözlü, Y., & Kutlu, Y. (2023). Leaf Image Classification Based on Pre-trained Convolutional Neural Network Models. *Natural and Engineering Sciences, 8*(3), 214-232.

[5] Choi, J., & Zhang, X. (2022). Classifications of restricted web streaming contents based on convolutional neural network and long short-term memory (CNN-LSTM). *Journal of Internet Services and Information Security (JISIS), 12*(3), 49-62.

[6] Chu, E.W., & Karr, J.R. (2017). Environmental impact: Concept, consequences, measurement. *Reference Module in Life Sciences*. 10.1016/B978-0-12-809633-8.02380-3

[7] Díaz-Uriarte, R., & Alvarez de Andrés, S. (2006). Gene selection and classification of microarray data using random forest. *BMC bioinformatics, 7*, 1-13.

[8] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation, 9*(8), 1735-1780.

[9] Jin, N., Zeng, Y., Yan, K., & Ji, Z. (2021). Multivariate air quality forecasting with nested long short term memory neural network. *IEEE Transactions on Industrial Informatics, 17*(12), 8514-8522.

[10] Juma, J., Mdodo, R.M., & Gichoya, D. (2023). Multiplier Design using Machine Learning Alogorithms for Energy Efficiency. *Journal of VLSI Circuits and Systems*, *5*(1), 28-34.

[11] Jurado, X., Reiminger, N., Benmoussa, M., Vazquez, J., & Wemmert, C. (2022). Deep learning methods evaluation to predict air quality based on Computational Fluid Dynamics. *Expert Systems with Applications*, *203*, 117294. https://doi.org/10.1016/j.eswa.2022.117294.

[12] Kumar, D. (2018). Evolving Differential evolution method with random forest for prediction of Air Pollution. *Procedia computer science*, *132*, 824-833.

[13] Kumaran, U., Radha Rammohan, S., Nagarajan, S. M., & Prathik, A. (2021). Fusion of mel and gammatone frequency cepstral coefficients for speech emotion recognition using deep C-RNN. *International Journal of Speech Technology, 24*(2), 303-314.

[14] Li, T., Hua, M., & Wu, X.U. (2020). A hybrid CNN-LSTM model for forecasting particulate matter (PM2. 5). *IEEE Access, 8*, 26933-26940.

[15] Muralidharan, J. (2020). A Air Cavity Based Multi Frequency Resonator for Remote Correspondence Applications. *National Journal of Antennas and Propagation (NJAP)*, 2(2), 21-26.

[16] Niranjan, D.K., & Rakesh, N. (2020). Real time analysis of air pollution prediction using IoT. *In Second international conference on inventive research in Computing applications (ICIRCA)*, 904-909.

[17] Niranjan, D.K., & Rakesh, N. (2021). Design of a Water and Oxygen Generator from Atmospheric Pollutant Air Using Internet of Things. In *Intelligent Data Communication Technologies and Internet of Things: Proceedings of ICICI 2020*, 361-375.

[18] Norazian, M.N., Shukri, Y.A., Azam, R.N., & Al Bakri, A.M.M. (2008). Estimation of missing values in air pollution data using single imputation techniques. *Science Asia, 34*(3), 341-345.

[19] Prasad, A.M., Iverson, L.R., & Liaw, A. (2006). Newer classification and regression tree techniques: bagging and random forests for ecological prediction. *Ecosystems, 9*, 181-199.

[20] Pushpavalli, R., Mageshvaran, K., Anbarasu, N., & Chandru, B. (2024). Smart Sensor Infrastructure for Environmental Air Quality Monitoring. *International Journal of Communication and Computer Technologies (IJCCTS), 12*(1), 33-37.

[21] Rakesh, N., & Kumaran, U. (2021). Performance Analysis of Water Quality Monitoring System in IoT Using Machine Learning Techniques. *In International Conference on Forensics, Analytics, Big Data, Security (FABS), 1*, 1-6.

[22] Ram, A., & Chakraborty, S. K. (2024). Analysis of Software-Defined Networking (SDN) Performance in Wired and Wireless Networks Across Various Topologies, Including Single, Linear, and Tree Structures. *Indian Journal of Information Sources and Services*, *14*(1), 39–50.

[23] Sethi, J.K., & Mittal, M. (2020). Analysis of air quality using univariate and multivariate time series models. *In 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 823-827.

[24] Setiawan, R.T., & Setiawan, E.B. (2023). The Sentiment Analysis of BBCA Stock Price on Twitter Data Using LSTM and Genetic Algorithm Optimization. *Synchronous: informatics engineering journal and research*, *8*(4), 2479-2489.

[25] Shakir, M., & Rakesh, N. (2018). Investigation on air pollutant data sets using data mining tool. *In 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC) I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 480-485.

[26] Sharanyaa, S., Vijayalakshmi, S., Therasa, M., Kumaran, U., & Deepika, R. (2022). DCNET: A Novel Implementation of Gastric Cancer Detection System through Deep Learning Convolution Networks. *In International Conference on Advanced Computing Technologies and Applications (ICACTA)*, 1-5.

[27] Stekhoven, D.J., & Bühlmann, P. (2012). MissForest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, *28*(1), 112-118.

[28] Yan, R., Liao, J., Yang, J., Sun, W., Nong, M., & Li, F. (2021). Multi-hour and multi-site air quality index forecasting in Beijing using CNN, LSTM, CNN-LSTM, and spatiotemporal clustering. Expert Systems with Applications, 169, 114513. https://doi.org/10.1016/j.eswa.2020.114513.

## Authors Biography

**Mohamed Shakir** received the bachelor's degree from Bangalore University, in 2001, the master's degree from Visvesvaraya Technological University, Belagavi, India, in 2011, and currently pursuing the Ph.D. degree from Amrita School of Computing, Bengaluru, Amrita Vishwa Vidyapeetham, India. He is currently a Full Assistant Professor with the School of Computer Science and Engineering, Presidency University, Bengaluru, India. Prior to this position, he was an Assistant Professor and HOD with the KNS Institute of Technology, Bengaluru, India. His area of interest is Deep Learning. Handled courses like Operating Systems, Data Structures, Design and Analysis of Algorithms and Programming Languages and other courses during recent years.

**U. Kumaran** (Member, IEEE) received the master's and Bachelor's degrees in computer science and engineering from Arunai Engineering College, affiliated to Anna University, Chennai, India, and the Ph.D. degree in computer science from the Vellore Institute of Technology, Vellore, India in June 2020. He is currently working as an Assistant Professor Selection Grade with Computer Science and Engineering Department, Amrita School of Computing, Bengaluru, Amrita Vishwa Vidyapeetham, India. He has more than 15 years of teaching experience and 5 years of research experience in the domain of computer science and engineering. He is currently guiding five PhD research scholars and his main research interests include Machine Learning, Cybersecurity, Internet of Things, Cloud computing and Privacy Preserving issues and data mining.

**Dr.N. Rakesh** working as Associate Professor in the Department of Information Science and Engineering at BMS Institute of Technology and Management, Bengaluru. He completed his B.E. in Information Science & Engineering from VTU, Belgaum; Karnataka in the year 2005. He holds 3 Master's degrees in M.Tech in CSE, MBA in Information Systems & M.Sc. in Computer Science from reputed colleges. Dr. Rakesh completed his Ph.D. in Computer Science & Engineering from PRIST University, Tanjore, Tamil Nādu in the year 2013 under the guidance of Honourable Professor Dr.S.K. Srivatsa IISc, Bangalore, and Retired Professor from Madras Institute of Technology, Guindy, Chennai. He has 18 years plus of teaching along with research experience. His area of interest includes Computer Networks, VoIP security protocols, Virtual private networks, Wireless Communication, Wireless sensor Networks, the Internet of Things, Wireless Channel Modelling, Wireless Communication, and also exploring Machine Learning, and Deep Learning areas. He has published 54 papers in various peered International Journals, International Conferences, and Springer Book chapter series, the majority of them in the Scopus index database. Also, he served as General chair, Session chair, Technical Program Chair, and International Program Chair for various international conferences in India and abroad. Designated Reviewer for various International Journals and Conferences. He is a Life member of the Computer Society of India, ACM – Professional Member, and IEEE – Professional Member.