Global NEST Journal, Vol 26, No 3, 05666
Copyright© 2024 Global NEST
Printed in Greece. All rights reserved

A real-time environmental air pollution predictor model using dense deep learning approach in IoT infrastructure

P. Preethi[1]*, T. Saravanan[2], R. Mohanraj[3] and PG. Gayathri[4]

[1]Department of Computer Science and Engineering, Kongunadu College of Engineering and Technology, Trichy

[2]Department of Computer Science and Engineering, GITAM University, Bengaluru, Karnataka, India

[3]Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning) Sri Venkateshwara College of Engineering and Technology Chittoor, Andhra Pradesh, India

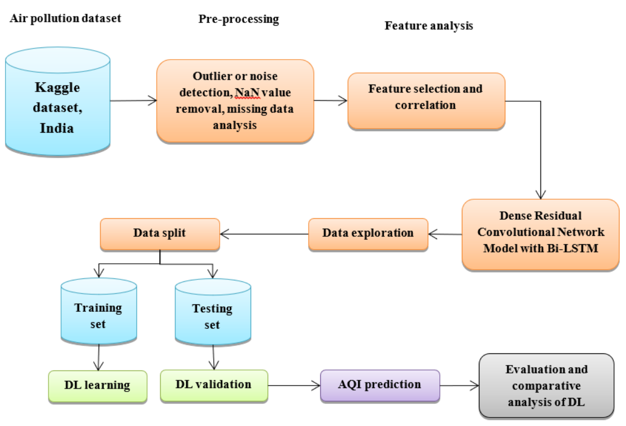
[4]Department of Artificial Intelligence and Data Science, Kongunadu College of Engineering and Technology, Trichy

Received: 18/12/2023, Accepted: 05/01/2024, Available online: 13/01/2024

*to whom all correspondence should be addressed: e-mail: preethi1.infotech@gmail.com

https://doi.org/10.30955/gnj.005666

Graphical abstract


Air pollution dataset
Pre-processing
Feature analysis
Kaggle dataset, India
Outlier or noise detection, NaN value removal, missing data analysis
Feature selection and correlation
Dense Residual Convolutional Network Model with Bi-LSTM
Data exploration
Data split
Training set
Testing set
DL learning
DL validation
AQI prediction
Evaluation and comparative analysis of DL

Abstract

With the technical advancements in Deep Learning (DL), it is probable to construct the predictor model for monitoring and controlling pollution from real-time data. Here, IoT techniques are used for sensing the emission rate from various factors and the predictor model is constructed using the available data, for instance, carbon monoxide prediction. Modern sensors are embedded to evaluate the level of pollutants and using these modern techniques, the source of emission rate is identified and notified to the specific environment. Deep learning concepts are used for predicting the pollution level based on the current and previous data attained from the sensors. Here, we have implemented a learning solution to predict carbon monoxide concentration hourly using the novel Dense Residual Convolutional Network Model with Bi-LSTM (Bidirection-Long Short Term Memory) with the spatial and temporal features by integrating the features of the present and previous pollutant data. The side output from the residual network model is used to evaluate prediction quality. The performance is compared with existing approaches like standard LSTM, CNN, pre-trained network model, etc. The experimentation is done in a Python environment, and the proposed model facilitates more prediction accuracy for the pollutants CO, SO_2, O_3 and NO_2 than other conventional network models and establishes a better trade-off.

Keywords: pollution, sensors, deep learning, residual network, the dense network, spatial and temporal features.

1. Introduction

A higher standard of life is promised by urbanization at the deterioration expense of air quality and the environment. The comprehensive requirement of fossil fuel-powered cars and machines in all places provides a considerable volume of hazardous gases and particular matter to the air. The critical component of life is air on Earth for plants, humans, animals, etc. The air pollution undermines the well-being, and the living creatures are developed directly. The air quality is declined very fast due to the sudden increase in urbanization. The various kinds of air pollutants are nitrogen oxides NOx (NO–NO2), carbon oxides COx (CO–CO2), Atmospheric Particulate Matter (PM for short) of diameters is equal to or lesser than 10 µm (PM10), sulphur oxides SOx (SO2, SO3, SO4), and the diameter PM is equal to or lesser than the 2.5 µm (PM2.5). The researchers are particularly interested in identifying and anticipating pollutants in real time [Lionetto *et al.* 2019; Gollakota *et al.* 2020; Department of Economic and Social Affairs, 2018].

The policies and the standards are defined by many countries worldwide for generating alerts to the citizens and observing the air pollution [Rodríguez-Urrego and Rodríguez-Urrego, 2020]. Moreover, the essential observations are done for the outdoor environment, and most measurements obtain static and average report values. Nevertheless, air quality differs in real-time, affecting many factors [Lionetto *et al.* 2019]. The speed of the wind, population density, distribution of pollutants, the location is outdoor or indoor, and different circumstances related to meteorological.

A mixture of gases and particles is concerned with air pollution, which has a higher concentration than the mentioned level of safety emitted to the environment [Palanisamy *et al.* 2023]. The pollutants sources are divided into two important sectors. The atmosphere particles are encompassed by the solid and liquid granular that suspends in the environment. They are (i) anthropogenic, artificial, and (ii) natural. The natural sources of pollution represent the natural triggered incidents that have the emission of harmful substances and destructive impacts on the environment. The natural occurrence is the outbursts of volcanos, and forest conflagrations, to generate more volume of air pollutants like NOx, COx, and SOx are examples. Meanwhile, several man-made causes exist, such as emissions from vehicles and the combustion of fuel that are deemed leading sources of air pollution. The pollutants have resulted in a specific matter, metal compounds, hydrogen, sulphur, ozone, and nitrogen.

Presently, the attention of research is focused on improving air quality and controlling air pollution [Punarselvam *et al.* 2023] and [Punarselvam *et al.* 2021]. The accurate approaches are developed, and the tools are made sure to ensure air quality. The critical part is prediction to obtain the aim. The process of prediction or forecast is an essential task in the research field of machine learning that can reduce the difference of the object's situation relative to the gathered information. Forecasting pollution is the concentration of pollutants projected in the long period or the short period. Since the 1960s, the study of air pollution has the control which is evolved. The awareness is increased, which leads to the evolution of the population regarding the devastating impact of the problem. Hence, the focus of the research is shifted to the forecast of air pollution. Forecasting on air pollution is divided into three classifications based on how the process of prediction is performed. They are (i) statistical models, (ii) potential forecasts, and (iii) numerical models. However, there are only two kinds of classifications depending on the forecast. They are (i) forecast concentration and (ii) forecasting pollution potentially [Punarselvam *et al.* 2021].

Statistical techniques and numerical modelling are utilized to forecast the concentration of pollutants. Nonetheless, the forecast can tell the ability and capacity potentially of the factors related to meteorological like the speed of the wind, and temperature, with other factors for diffusion or dilution of the pollutants of air. A warning is provided if the conditions weather are matched the standards for severe pollution, which is possible. Forecasting is the first tool for potentially predicting air quality [Kok *et al.* 2017]. The forecast has the concentration that directly indicates the concentration of pollutants in the particular region, and the quantitative values are forecasted. The quality of air is predicted that requires meteorological features than the concentration of pollutants is best to expect the concentrations in future. Moreover, the different sources produce the data in [Li *et al.* 2017] that has images, satellites, and calculated data from the ground station merged to predict better. In computer science and statistics, Linear machine learning (ML) models are used to solve the issue of prediction in the data-driven technique essential during the many linear regression [Hernandez *et al.* 2020]. Moreover, the behaviour of air pollutants is nonlinear. Hence the SVR (Support Vector Regression) is required [Ventura *et al.* 2019]. Nevertheless, the present research provides methods based on deep learning as the most accurate method to predict the pollutants in the air [Roser, 2023; Harrow *et al.* 2020]. Hence, many deep learning-based and nonlinear algorithms are required in the proposed system for the PM2.5 prediction with the help of gathered data during the last 24 hours for the next hour. However the available works, does not fulfil the requirements of the researchers. This work intends to resolve these issues and the important contributions are presented below:

Air pollutants need to be predicted using the online available Kaggle dataset, which is measured for four cities like Chennai, Delhi, Mumbai and Hyderabad. The pre-processing step is adopted for handling the missing values, NaN values and outliers.

The better multivariate pollutant features are considered, and the input variable is chosen with the help of the correlated data model. The input variables perform the essential part in the forecasting model performance.

The theoretical application for forecasting using the proposed Dense Residual Convolutional Network Model with Bi-LSTM (Long Short Term Memory) is suggested for predicting NO2, O3, SO2 and O3. The anticipated model is appliedover the parameters of BiLSTM to optimize the proposed system.

The suggested application related to forecasting is verified on the real datasets successfully. Also, the comparison is made with the model of predicting and the adequate standard models like GRU, BiGRU, LSTM, BiLSTM, ConvLSTM, VAE, A-GRU, A-LSTM and A-VAE.

The proposed prediction application has the best benefit compared with the models demonstrated in the present research using deep learning methods. Specifically, optimization algorithms and the deep learning models in the proposed system are used, and it is integrated with the mutual data method for selecting the model inputs with the best features.

The remainder section provides the essential things required for this study and initiates the Dense Residual Convolutional Network Model with Bi-LSTM (Long Short Term Memory)-based prediction methodology. Sections 2 and 3 provide a detailed analysis of the existing and proposed methods. The results associated with the proposed model are shown in section 4 and the summary in section 5.

2. Survey of related works

The importance of forecasting atmospheric pollutants is essential in this epidemic era, hence researchers looked into methods for forecasting Particulate Matter (PM) concentration as accurately and in advance. But, applying these methodologies in the real time necessitated the development of models that could collect actual environmental signals and utilise Machine Learning (ML) algorithms to anticipate the subsequent level of pollution. [Huang and Kuo, 2018] introduced APNet, which combines ordinary LSTM with CNN to better forecast PM2.5 in the environment like smart city. They used the information in [Liang *et al.* 2015] to predict the following hour using PM2.5 concentration data from the previous 24 hours, as well as accumulated hours of rainfall and accumulated wind velocity. Their idea outperformed LSTM, CNN, and other machine learning algorithms individually. They used the Root Mean Square Errors (RMSEs), Mean Absolute Errors (MAEs), Index of Agreements (IAs), and the Pearson correlation coefficient (PCC) to analyse their idea. They experimented with their approach to see if it was feasible and practicable to anticipate PM2.5. Even so, because of the source of PM2.5 pollution is variable, algorithm predictions did not precisely track the real trend, which was displaced and disordered.

[Yang *et al.* 2020] proposed an integrated deep learning model that combined LSTM with CNN plus LSTM integrated Gated Recurrent Unit to enhance the estimation of PM2.5 and 10 variant for the following 7 days (168 hours), respectively. RMSE and MAE were used to evaluate their experiments. For five random places, their hybrid methods done much better than other single approaches. For PM2.5 and PM10, CNN-LSTM and CNN-GRU were better fitted, respectively. These hybrid systems, on the other hand, could only estimate the future peak and lowest values of PM2.5. An assessment of four ML methods (LSTM, Support Vector Regression, Special Tree structures, and Random Forest (RF)) was also published in [Moursi *et al.* 2019]. They forecasted the following hour based on the previous 48 hours. The variety of machine learning techniques compared in the investigation was limited. For most techniques, there was a slight difference between actual and anticipated values. The Special Trees algorithm was determined to have the better predictive outcomes in terms of RMSE measure and R2 determination coefficient range.

[Li *et al.* 2020] created a hybrid Deep Learning (DL) multivariate CNN+LSTM method to forecast PM2.5 intensity for the upcoming 24 hours in Beijing city by utilizing the data from previous 7 days whose dataset taken as similar in [Liang *et al.* 2015]. LSTM may well do prediction by utilising long-term historical data input, while CNN might retrieve air quality features, reducing time of training needed. They compared CNN-univariate LSTM's and multivariate variants to an LSTM-only variant. RMSE & MAE were utilised to assess their job. However, further assessment measures, such as R2 or IoA, that indicate similarity to real values than just other error metrics and it have been utilised to confirm their system efficiency.

[Biancofore *et al.* 2017] used meteorological factors and PM10 records in three settings for comparison reasons to detect the daily mean concentration rate of PM10 between 1 - 3 days in advance. A multivariate Linear Regression (LR) model and a Neural Network (NN) with iterative as well as non-recursive structures were used in the experiments. Carbon-monoxide (CO) was also added as the input factor, which improved the prediction's effectiveness. Finally, without the need for a past data of PM2.5, the percentage of the same was predicted along with meteorological sample sets, CO and PM10. As measurement methods, they employed the Coefficient of Correlation (CC), Normalized Mean Squared Errors (NMSEs), Fractional Biasing (FB), and the Factor of 2 (FA2). In all the existing studies, the recursion based NN model gives better trade-off. More machine learning techniques, on the other hand, may have been utilised further to test their methods. Spatio-temporal methods were used in some of the studies to address the lagging of air quality assessment tools in every area. These technologies forecast quality of air at a specific time and location based on the data from another source.

The same method is used to improve prediction at a specific place based on data gathered nearby. [Luo *et al.* 2019] developed a method that used PM_x, and O3 data to forecast quality of air for the following 2 days based on the 3 days record of attributes for each surveillance stations in the London city as well as Beijing city. Gated-DNN, Seq2Seq and Light-GBM were developed to create global and regional air quality features. LightGBM was used to pick features, whereas Gated-DNN collected spatially and temporally correlation ranges, and the Seq2Seq featured an encoding model that summarised past recorded attributes and also a decoding that added forecasted meteorological factor inputs for improvising the accuracy metric. The combined performance of all the techniques (AccuAir) outperformed the separate components. Symmetric Mean-Absolute-Percentage Errors (SMAP) was used to assess their models. Although LSTM has been shown to be particularly proficient in time series forecasting, they did not consider it in their Seq2Seq system. All the existing methods are compared in Table 1.

2.1. Research gap and significance

Based on the findings of the literature survey section, it is expected that existing statistical models have certain limitations, and the suggested technique interprets the unique ideology to bridge the gap. The following points demonstrate the gap as well as the importance of the suggested research

The existing approaches concentrate only on a univariate climate which affects the performance in terms of prediction error like RMSE, MAPE, and MAE. But the proposed CNN+BiLSTM mechanism works well even in the case of a multivariate environment which substantially reduces the errors and is not achieved in the existing standard approaches like LSTM, GRU, standard BiLSTM, A-LSTM, etc.

The existing methods fail to enhance the pollution prediction in both the univariate and multivariate temporal time-series data with single (structure) source data set. Historical data from the target station and nearby stations along with meteorological feature is combined with other factors and included into the model in our research. According to the findings, the proposed combination is more effective than others in extracting spatiotemporal characteristics and performing Fine particulate matter prediction accuracy.

The correlation among the features is not analyzed by the existing methods that need to be addressed in the proposed model with the dense CNN network model. The correlation mapping among the features helps enhance the prediction accuracy in proposed work by the extraction of data's spatial and temporal features. It is achieved by combining the benefits of the CNN with Bi-LSTM, which is effective at filtering out the spatial and temporal characteristics of data between pollutant elements and weather, as well as between different adjacent stations.

Table 1. Comparison of the existing approaches

Ref. no.	Technique applied	Parameters considered for evaluation	Forecasting/predicting period	Strength	Shortcomings
[Huang and Kuo, 2018]	APNet	RMSE, MAE and IA	5 hours	The feasibility and usefulness of their idea for forecasting PM2.5 was tested experimentally.	Algorithmic forecasts were a little distorted and chaotic, so did not accurately match the real trend
[Yang *et al.* 2020]	HYBRID DL	RMSE and MAE	7 days	For PM10, 2.5, CNN-GRU and LSTM performed better.	Hybrid algorithms only had an unstable prediction of future PM2.5 maximum and with lowest concentrations
[Moursi *et al.* 2019]	SVR, LSTM, RF and ST	RMSE and R2	1 hour	After experimenting with a variety of methods, it was discovered that Special Tree algorithms provides the best results	The number of ML algorithms examined in the study was restricted. For most techniques, there was a slight gap between the actual and anticipated
[Li *et al.* 2020]	CNN+LSTM	MAE and RMSE	24 hours	LSTM perform prediction by utilising longterm past recorded input data, whereas CNN could retrieve air quality attributes and so reducing the time for training.	More assessment parameters, such as R2 as well as IA, indicating similarity to real values, instead of errors measures and may have been utilised to validate their methods' performance
[Biancofore *et al.* 2017]	Multiple Regression	FA2, NMSE and FB	1-3 days	Without a record of PM2.5, the concentration of PM2.5 was forecasted with meteorological factors, as well as PM10 and CO	To validate their methods further, more ML models may have been deployed
[Luo *et al.* 2019]	Spatio-temporal method	SMAPE	48 hours	The combination of the three concepts outperformed the individual elements examined	LSTM was not used in the Seq2Seq model, despite the fact that it has not been demonstrated to be very effective in time series forecasting

The prediction ability of the existing supervised approaches is not satisfying, and it gives a complex network model with a time complexity. Bidirectional ability of LSTM can shorten the training time. Thereby, decreased time complexity based on logarithmic asymptotic functions.

3. Methodology

This section briefly describes the proposed methodology concept and the preliminary requirements for air pollutants prediction. The proposed Dense Residual Convolutional Network Model with Bi-LSTM is introduced, and the overall schematic representation is presented in Figure 1.

3.1. Dataset description

Air is what helps humans to live. Understanding and monitoring its quality is highly solicited to human well-being. This research considers the Kaggle dataset (https://www.kaggle.com/datasets/rohanrao/air-quality-data-in-india) for predicting air quality data in India (2015-2020). The dataset comprises air quality data and AQI daily and hourly from various stations across multiple cities. The cities like Visakhapatnam, Talcher, Patna, Thiruvananthapuram, Shillong, Lucknow, Mumbai, Kochi, Jaipur, Kolkata, Kochi, Jorapakhar, Hyderabad, Gurugram, Guwahati, Delhi, Ernakulam, Coimbatore, Chandigarh, Chennai, Bhopal, Bengaluru, Amaravathi, Ahmedabad, Aizawl, Amristar, and Brajrajnagar are considered for air quality monitoring. The dataset is online accessible, which can be acquired directly from the above link.

3.2. Data acquisition

Firstly, the data quality is essential to visualize efficiently and create the learning model effectively. The noise in the dataset is reduced in the pre-processing steps that improve the processing speed eventually and generalization ability of the learning approaches. The missing data and the outliers are the common errors during data extraction and monitoring application. The data pre-processing step carries out different works on the data like changing or removing the data outlier, filling out not-a-number (NAN) value, etc. Among all the other features, Xylene is the most missing value, and CO attains the less missing values. Because of the various factors like a station can identify the data yet does not possess the devices for recording it, many missing values exist.Complete missing values are filled and median values over every feature for resolving the issue of missing data. Then, normalization is used for standardizing the data to ensure that the consideration of variables is not affected using the units or ranges. The process of data normalization requires bringing the various data attributes to the same range of measurement. An important role is played by this process in the ML models' stable training, and the performance is boosted.

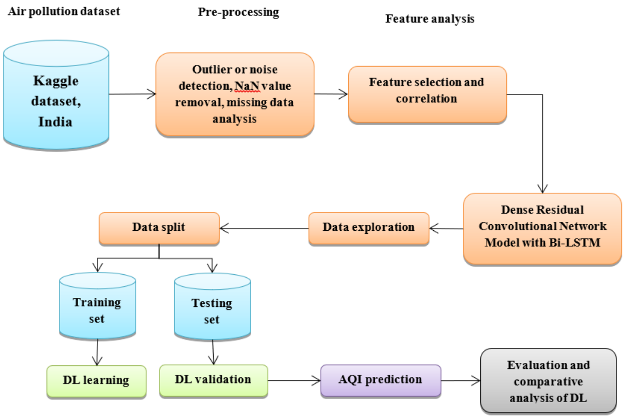

Air pollution dataset
Pre-processing
Feature analysis
Kaggle dataset, India
Outlier or noise detection, NaN value removal, missing data analysis
Feature selection and correlation
Dense Residual Convolutional Network Model with Bi-LSTM
Data exploration
Data split
Training set
Testing set
DL learning
DL validation
AQI prediction
Evaluation and comparative analysis of DL

Figure 1. Block diagram of the proposed model

3.3. Convolution neural network for feature extraction

Here, 1D signal with sub-sampling are adopted to execute attribute mapping. There are diverse convolutional and pooling levels that deal with 1D convolution patches which perform feature extraction and weights in the dense network environment. The weight contributes to the diminishment of diverse training parameters where the feature maps extract the discriminant features from multiple input vectors via convolution filters.

Convolution layer: This layer extracts data from the database using multiple convolution filters. The local linkfacilitates the preservation of the resources compared to the entire connection. The feature extraction processes via sub-sampling and convolution operation are provided in Figure 2. The expression for data mapping to the convolution layers via diverse feature filters f_x from different segments is considered, and the output features are provided through the nonlinear transformation based on multiple kernels:

$$x_j^l = f\left(\sum_{i\varsigma M_j} x_i^{l-1} * k_{ij}^l + b_j^l\right) \tag{1}$$

Here, M_j specifies the input feature, l determines the number of layers, b_j identifies bias, k sets convolution kernel, and the activation function f includes both sigmoid and hyperbolic tangent functions.

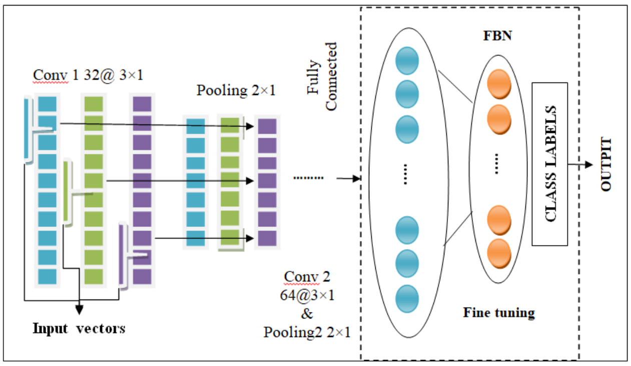


Figure 2. 1D CNN model

Pooling layer: Here, sub-sampling is adopted to attain the lower resolution feature maps. The maximal sampling operation reduces the generated feature maps to one half of the last layer.

Fully connected layer: Usually, it is provided after the output layer to transfer description for categorization. The proposed CNN model is developed with essential layer data (training phase) and gradient descent is represented for parameter updation. It executes the classification using the take outfeatures. The pooling layer maps the output from the input representation as in Eq. (2):

$$S_j^l = \max\left(a_i^l\right),\ i\varsigma R_j^l \tag{2}$$

Here, Rj^l specifies the j^{th} pooling domain in the l^{th} layer, and a_i^l represents pooling layer (index i)features. The kernel size is s*1, reducing the output to 1/s of its input. The attribute maps f_v is take out by the 1D CNN, which is fed into the LSTM as follows:

$$O = f\left(b_0 + w_o f_v\right) \tag{3}$$

Here, b_o and w_o specifies bias and weight. The parameter optimization is initiated with error computation using output y from the classification level and information layer i Let m defines m^{th} layer (CNN) where m = 1, …, M, K_M determines class labels. The mean-squared error (output layer) relating to the i^{th} input vector is expressed as in Eq. (4):

$$E_i = MSE\left(l_j^i,\left[y_1^M,\ldots,y_{K_M}^M\right]\right)=\sum_{j=1}^{K_M}\left(y_j^M-l_j^i\right)^2 \tag{4}$$

Parameters are optimized by reducing the classification error. Here, l_j^i specifies the class label in the final layer (j^{th} neuron); $\left[y_1^M,\ldots,y_{K_M}^M\right]$ specifies vectors. The MSE derivative determines weight w_m^{m-1} and b_n^m determines bias, and the optimizer is selected to diminish the error iteratively. Therefore, the bias and weight of the prior layer associated with the neuron are updated using delta Δ_n^m over the m^{th} layer is:

$$\frac{\partial E}{\partial w_{j_n}^{m-1}}=\Delta_n^m y_j^{m-1},\frac{\partial E}{\partial b_n^m}=\Delta_n^m \tag{5}$$

The back propagation is executed from the initial layer to the final as:

$$\frac{\partial E}{\partial s_n^m}=\Delta s_n^m=\sum_{j=1}^{k_{m+1}}\frac{\partial E}{\partial x_j^{m+1}}\frac{\partial x_j^{m+1}}{\partial s_n^m}=\sum_{j=1}^{k_{m+1}}\Delta_j^{m+1}w_{nj}^m \tag{6}$$

Here, us_n^m ing a specified zero-up sampling map, Δ_n^m is shown as:

$$\Delta_n^m=\frac{\partial E}{\partial y_n^m}\frac{\partial y_n^m}{\partial x_n^m}=\frac{\partial E}{\partial us_n^m}\frac{\partial us_n^m}{\partial y_n^m}f'\left(x_n^m\right) \tag{7}$$
$$=up\left(\Delta s_n^m\right)\beta f'\left(x_n^m\right)$$

Here, $\beta = (SS)^{-1}$ specifies the number of y_n^m elements, and it is averaged to attain S_n^m elements, and it is expressed as:

$$\Delta s_n^m=\sum_{j=1}^{k_{m+1}}conv1Dz\left(\Delta_m^{m+1},rev\left(w_{nj}^m\right)\right) \tag{8}$$

Here, rev(.) specifies array and con1D(.) specifies fully connected operation. The optimization parameter is provided below:

```
Algorithm: 1D CNN Parameter optimization
Input: training and testing input (x and x̂) and class label and testing label (y and ŷ)
1. Bias and weight initialization over the dense network environment;
2. For successive iterations, D
i)Attain layer-wise output, $y_j^m, for\,all\, j\in\left[1,k_m\right] and\, all\, m\in\left[1,m\right]$
ii) Evaluate $\Delta_n^m,\Delta_n^m, for\,all\, n\in\left[1,K_m\right] and\, m\in\left[2,M-1\right]$
iii) Update every bias and weight with the provided learning rate ε
$w_{jn}^{m-1}\left(t+1\right)=w_{jn}^{m-1}\left(t\right)-\varepsilon\frac{\partial E}{\partial w_{jn}^{m-1}}$
$w_{jn}^{m-1}\left(t+1\right)=w_{jn}^{m-1}\left(t\right)-\varepsilon\frac{\partial E}{\partial w_{jn}^{m-1}}$
3. End for
Output: 1DCNN model-based feature extraction
```

3.4. Long-short term memory

It eliminates the gradient vanishing and explosion during training process where input, output, and forget gates are introduced. These gates help in resolving the issues associated with the training problem. Here, x_t specifies the LSTM's time series value l^{th}, $c_t$defines the memory cell that intends to manage time information transformation (pollutant-based analysis). The input data is used for determining the information from the present time to the successive time; while the forget gate specifies the information retained by the present time (acquired from the past) and the output gate specifies the current state output to the successive state which is expressed as:

$$i_t=\sigma\left(W_{xi}x_t+W_{hi}h_{t-1}+b_i\right) \tag{9}$$

$$f_t=\sigma\left(W_{xf}x_t+W_{hf}h_{t-1}+b_f\right) \tag{10}$$

$$o_t=\sigma\left(W_{xo}x_t+W_{ho}h_{t-1}+b_0\right) \tag{11}$$

$$\tilde{c}_t=\tan h\left(W_{xc}x_t+W_{hc}h_{t-1}+b_c\right) \tag{12}$$

$$c_t=f_t e c_{t-1}+i_t e\tilde{c}_t \tag{13}$$

$$h_t=o_t e tan\,h\left(c_t\right) \tag{14}$$

3.5. Bidirectional LSTM

It combines backwards and forwards LSTM that fits the data in backward and forward directions. It helps in establishing the prediction with the concatenation process. The drawback of the existing LSTM relies on one direction time-based analysis; however, BiLSTM performs a reverse function where the data patterns are analyzed in a reverse process, which is not achieved in LSTM. Figure 3 depicts the forward and reverse the direction of biLSTM where *S* and S′ represents the time-series information and L_i and L_i^I specifies the reverse function.

3.6. Layer normalization

When the model is proposed deeply or densely, it has many problems. A small modification in a single parameter of one layer leads to constant modification in the successive layers associated with the parameter. Therefore, the training efficiency leads to a huge reduction. The layers' output passes via the activation function (exceeds the range) causes the neuron functionality failure. To handle these issues, layer normalization is required, which is expressed as:

$$\mu_B=\frac{1}{m}\sum_{i=1}^{m}x_i \tag{15}$$

$$\sigma_B^2=\frac{1}{m}\sum_{i=1}^{m}\left(x_i-\mu_B\right)^2 \tag{16}$$

$$\hat{x}_i=\frac{x_i-\mu_B}{\sqrt{\sigma_B^2+\varepsilon}} \tag{17}$$

$$y_i=\gamma\hat{x}_i+\beta \tag{18}$$

Here, x_i specifies input value and y_i specifies output (layer normalization); *m* specifies the number of inputs (mini-batch); μ_B specifies mean (input); σ^2_B specifies average input variance, and $\hat{x}_i$ selects normalized x_i Parameters like γ and β are learned by back-propagation. After normalization, output is constrained to a specific range and training process is enhanced efficiency.

3.7. Attention mechanism

The attention model helps project the significance of various features, filters out the higher value information and eliminates irrelevant data. The prediction is made by allocating the features with weight, and it works effectually in longer time series. The attention block is placed between the LSTM and convolutional layer, where the attention block output is shown in Eq. (19), and it learns the weight automatically as:

$$c_i = \sum_{j=1}^{L\_x} \propto_{ij} h_j \tag{19}$$

Here, h_j specifies global features, α_{ij} specifies features allocated with weight using the attention block, c_i specifies the attention block's output. This mechanism performs soft and hard attention where the former mechanism weights the global features to concentrate on a specific region in a differential training process. At the same time, the latter model diminishes the training cost by choosing a particular region. But restricting the input information is not appropriate for handling the time-based issues and leads to a non-differentiable training process. Thus, the soft attention mechanism is adopted in this research.

3.8. Deep residual block

In section illustrates the residual block as depicted in Figure 3. The residual block comprises the BiLSTM layer, additional layers (dropout, ReLU, normalization layer and fully-connected layers), and intermediate connections to eliminate over-fitting and vanishing gradient issues. We have modelled a novel DenseNet and ResNet model with the structural level design where the intermediate blocks are examples of the dense residual block. With the provided *n* blocks, the output of the blocks (*l*−1) is specified as r_t^{l-1} for *t* position. Then, the dropout layer output is expressed as:

$$d_t^l = g^l\left(W_{fc} \cdot r_t^{l-1}\right) \tag{20}$$

Here, W_{fc}^l specifies the weighted matrix FC layer, and g^l specifies composite function (normalization), dropout and ReLU. d_t^l vector represents BiLSTM input and primary LSTM function is expressed as:

$$d_t^l = g^l\left(W_{fc} \cdot r_t^{l-1}\right) \tag{21}$$

$$\begin{bmatrix} \tilde{c}_t^l \\ o_t^l \\ i_t^l \\ f_t^l \end{bmatrix} = \begin{bmatrix} \tanh \\ \sigma \\ \sigma \\ \sigma \end{bmatrix} \left(W_p^l \begin{bmatrix} d_t^l \\ \tilde{h}_{t-1}^l \end{bmatrix} + b_p^l \right) \tag{22}$$

$$c_t^l = i_t^l \odot \tilde{c}_t^l \tag{23}$$

$$\tilde{h}_t^l = o_t^l \odot \tanh\left(c_t^l\right) \tag{24}$$

Here, $\tilde{c}_t^l$, i_t^l, o_t^l, f_t^l specify cell state and input, output and forget gate, W_p^l specifies weighted matrix, b_p^lspecifies bias matrix, σ specifies a sigmoid function, and ʘ specifies element-wise product. Here, backward LSTM is represented as $\overleftarrow{h}_t^l$, and $\vec{h}_t^l$ is forward LSTM. Then, these two hidden states are concatenated to get $h_t^l = [\vec{h}_t^l : \overleftarrow{h}_t^l$] vector. Then, the residual's *l* block final output is expressed as:

$$r_t^l = h_t^{l-1} + h_t^l \tag{25}$$

Therefore, a novel identity connection is provided to BiLSTM to construct a deep residual model. To define the deep residual model more specifically, the residual's *l* block is expressed as:

$$r_t^l = h_t^l + \mathrm{H}^l\left(r_t^{l-1}\right) \tag{26}$$

Here, H^l specifies the composite function of the residual's l block.

3.9. Proposed dense residual convolutional network model with Bi-LSTM

The proposed dense network structure presents the Dense Residual Convolutional Network Model with Bi-LSTM, as shown in Figure 3. The proposed model includes CNN, attention mechanism, LSTM and dense residual network model. The first layer extracts features of the pollutants, and the attention model allocates weight to the features. The proposed model helps in predicting the pollutants based on the feature information. The functionality is explained below:

Some effectual pollutant features are extracted using convolutional kernel striding such as Local Patterns, Spatial Hierarchies, Size and Shape of Pollutant Sources and Variations in Concentrations. 1) The proposed model is provided with 1D convolutional layer where the kernels are set as 16, 32 and 64, respectively, where the size is 2 and the striding step is 1. After every convolution layer, the max-pooling with a window size of 2 is set, and the striding is set as 1. The role of the level is to decrease the characteristic complexity and eliminate over-fitting issues.

2) By allocating the weights via the attention module can enhance the impact of features (temporal and spatial) are increased and reduces the consequences of the non-influencing features. Thus, the over-fitting and vanishing gradient issues are resolved.

3) The extracted features are provided to the LSTM and proposed BiLSTM (dense network structure) and help acquire the pollutants for successive days. The historical input data is chosen by experimenting with the fine-tuned multiple parameters, and the trained model gives higher prediction accuracy.

4. Numerical results

To evaluate the pollution prediction and compare the existing approaches, some metrics like Coefficient of determination (R^2), Root Mean Square Error (RMSE),Mean Absolute Error (MAE), Explained Variance (EV), Mean Absolute Percentage Error (MAPE),Relative Mean Bias (RMB), and Mean Bias Error (MBE) are used. The following are the expression of the provided metrics:

$$R^2 = \frac{\sum_{t=1}^{n}\left[\left(y_t, \bar{y}\right).\left(\hat{y}_t - \bar{y}\right)\right]^2}{\sqrt{\sum_{t=1}^{n}\left(y_t - \bar{y}\right)^2}.\sqrt{\sum_{t=1}^{n}\left(\hat{y}_t - \bar{y}\right)^2}} \tag{27}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}\left(y_t - \hat{y}_t\right)^2} \tag{28}$$

$$MAE = \frac{\sum_{t=1}^{n} \lvert y_t - \hat{y}_t)^2}{n} \tag{29}$$

$$MAPE = \frac{100}{n}\sum_{t=1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t}\right| \tag{30}$$

$$EV = 1 - \frac{var\left(\hat{y} - y\right)}{var\left(y\right)} \tag{31}$$

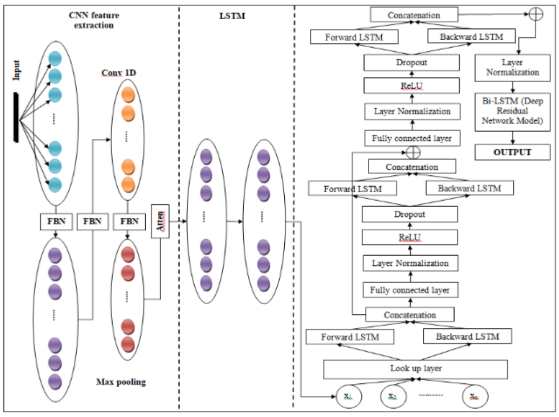


Figure 3 Proposed Dense Residual Convolutional Network Model with Bi-LSTM

$$MBE = \frac{1}{n}\sum_{t=1}^{n}\left(\hat{y}_t - y_t\right) \tag{32}$$

$$rMBE = \frac{\sum_{t=1}^{n}\left(\hat{y}_t - y_t\right)}{\sum_{t=1}^{n} y_t}.100 \tag{33}$$

4.1. Analysis

The experimentation intends to investigate the anticipated model performance in predicting the pollutant level (CO, SO_2, O_3 and NO_2) concentration. In the prediction phase, the proposed model is trained to extract the temporal dependencies based on time-series data of every pollutant. The target is to predict the provided measurement sequence's successive values (based on biLSTM). The model is implemented in Intel i7 process with 12GB RAM. Here, Python 2.7.0 is used where the training set is composed of the daily concentration level of four diverse pollutants (CO, SO_2, O_3 and NO_2). Here, *k*–fold cross-validation with K = 10 is used to construct the prediction model where the hyper-parameters are set for every model and named as optimizer → adam, loss function → cross entropy, batch size → 250, epochs → 100 and learning rate → 0.001. Table 2 depicts the value attained with validation metrics of provided testing data from the provided data. While evaluating the metrics, the proposed model is considered the best approach for assessing the prediction problem with fulfilling accuracy and higher efficiency. The proposed model outperforms various existing approaches, i.e. A-LSTM, ConvLSTM, GRU, BiGRU, LSTM and GRU-A, in predicting all experimented pollutants (CO, SO_2, O_3 and NO_2) are measured. As expected, the anticipated model attains lesser prediction errors (RMSE, MBE, MAE and RMSE)and a higher score of *EV* and R^2It is attributed with time-dependencies and chose appropriate features utilizing the attention model. The significance of the general approaches needs to be highlighted, and the Bidirectional RNN models like BiGRU and BiLSTM are superior to the unidirectional models (GRU and LSTM). It is owing to the capability of bidirectional approaches while processing data in backward and forward directions.

Table 2. Pollutant prediction based performance analysis

Pollutant	Model	RMSE	MAE	R2	EV	MAPE (%)	MBE	RMBE (%)
NO2	GRU	13.6	11.0	0.9	0.9	23	-4.6	-5.8
	LSTM	13.7	11.4	0.8	0.9	25	-5.5	-6.8
	BiGRU	11.4	9.3	0.9	0.94	20	-4.8	-6.1
	BiLSTM	10.9	8.9	0.9	0.94	19	-4.3	-5.5
	A-GRU	15.7	13	0.8	0.95	29	-8.7	-10.4
	A-LSTM	17.9	15.4	0.8	0.9	35	-9.7	-11.5
	VAE	9.4	7.1	0.9	0.87	14	-2.4	-3.2
	Proposed	8.5	6.2	0.95	0.956	12	-3.2	-3.1

O3	GRU	0.02	0.019	0.9	0.9	7	0.005	1.7
	LSTM	0.02	0.02	0.93	0.9	7	0.007	2.4
	BiGRU	0.02	0.018	0.018	0.9	7	0.005	1.7
	BiLSTM	0.022	0.018	0.017	0.94	7	-0.001	-0.3
	A-GRU	0.025	0.019	0.019	0.94	7	0.004	1.2
	A-LSTM	0.033	0.025	0.025	0.92	8	0.011	3.3
	VAE	0.02	0.018	0.018	0.95	7	0.0017	0.52
	Proposed	0.019	0.017	0.016	0.956	6	0.0015	0.45
SO2	GRU	1.125	0.8	0.8	0.86	14	-0.2	-2.9
	LSTM	1.243	0.9	0.83	0.83	15	-0.2	-2.3
	BiGRU	0.918	0.7	0.9	0.92	12	-0.3	-4.3
	BiLSTM	0.816	0.6	0.92	0.93	10	-0.12	-1.4
	A-GRU	1.8	1.5	0.62	0.63	24	-0.3	-3.8
	A-LSTM	1.9	1.6	0.58	0.59	24	-0.3	-3.45
	VAE	0.59	0.4	0.96	0.96	6	-0.0075	-0.84
	Proposed	0.45	0.356	0.973	0.972	5	-0.0065	0.75
CO	GRU	0.2	0.18	0.95	0.94	5	-0.065	-1.45
	LSTM	0.2	0.19	0.93	0.93	6	-0.03	-0.8
	BiGRU	0.26	0.22	0.93	0.95	7	-0.15	-3.4
	BiLSTM	0.23	0.19	0.94	0.96	6	-0.14	-3.3
	A-GRU	0.3	0.29	0.85	0.86	8	-0.07	-1.7
	A-LSTM	0.4	0.36	0.79	0.80	9	-0.10	-2.4
	VAE	0.18	0.13	0.96	0.96	4	0.012	0.30
	Proposed	0.15	0.10	0.97	0.98	2	0.010	0.25

Table 3. CO pollutant based performance analysis

State	Model	RMSE	MAE	R2	EV	MAPE (%)
Chennai	GRU	2.00E-02	1.56E-02	0.95	0.95	8.4
	LSTM	2.00E-02	1.56E-02	0.95	0.95	8.1
	BiGRU	1.73E-02	1.42E-02	0.95	0.95	7.3
	BiLSTM	2.00E-02	1.49E-02	0.94	0.94	7.8
	VAE	1.70E-02	1.51E-02	0.95	0.95	8.7
	A-GRU	2.00E-02	0.0153	0.94	0.94	8.9
	A-LSTM	0.24E-02	0.0185	0.91	0.91	10.8
	Proposed	1.51E-02	1.26E-02	0.96	0.96	7.1
Delhi	GRU	1.72E-02	1.12E-02	0.88	0.88	7.2
	LSTM	1.75E-02	1.22E-02	0.87	0.87	8.1
	BiGRU	1.20E-02	9.00E-03	0.92	0.93	5.4
	BiLSTM	1.20E-02	9.00E-03	0.92	0.92	5.1
	VAE	1.4E-02	1.00E-02	0.89	0.89	7.6
	A-GRU	1.75E-02	0.011	0.88	0.89	7.3
	A-LSTM	1.75E-02	0.012	0.87	0.87	8.2
	Proposed	1.10E-02	7.70E-03	0.94	0.94	4.7
Mumbai	GRU	3.61E-02	2.54E-02	0.78	0.83	12.5
	LSTM	3.46E-02	2.40E-02	0.80	0.85	12.20
	BiGRU	3.46E-02	2.30E-02	0.79	0.86	10.8
	BiLSTM	3.46E-02	2.29E-02	0.80	0.86	10.8
	VAE	3.38E-02	2.10E-02	0.82	0.82	14.02
	A-GRU	4.24E-02	0.031	0.70	0.82	15.28
	A-LSTM	2.88E-02	0.030	0.72	0.83	14.76
	Proposed	2.45E-02	1.95E-02	0.88	0.86	10.60
Hyderabad	GRU	2.45E-02	1.84E-02	0.80	0.82	23.5
	LSTM	2.45E-02	1.84E-02	0.79	0.80	23.6
	BiGRU	2.24E-02	1.72E-02	0.83	0.85	21.5
	BiLSTM	2.45E-02	1.78E-02	0.80	0.81	21.1
	VAE	2.50E-02	1.85E-02	0.78	0.79	27.2
	A-GRU	2.83E-02	0.02	0.75	0.77	28.3
	A-LSTM	2.83E-02	0.021	0.74	0.75	28.2
	Proposed	2.19E-02	1.56E-02	0.88	0.87	19.6

Table 4. NO_2pollutant based performance analysis

State	Model	RMSE	MAE	R2	EV	MAPE (%)
Chennai	GRU	3.32E-02	2.61E-02	0.90	0.91	27.68
	LSTM	3.00E-02	2.44E-02	0.91	0.92	27.24
	BiGRU	2.83E-02	2.36E-02	0.92	0.92	22.8
	BiLSTM	3.00E-02	2.35E-02	0.92	0.92	22.2
	VAE	3.00E-02	2.55E-02	0.92	0.94	25.1
	A-GRU	3.87E-02	0.032	0.86	0.86	32.7
	A-LSTM	4.24E-02	0.034	0.84	0.84	32.15
	Proposed	2.72E-02	2.14E-02	0.93	0.93	19.7
Delhi	GRU	2.240E-02	1.12E-02	0.90	0.90	24.2
	LSTM	2.45E-02	1.22E-02	0.87	0.91	32.3
	BiGRU	2.00E-02	9.00E-03	0.92	0.94	24.08
	BiLSTM	2.00E-02	9.00E-03	0.92	0.93	22.2
	VAE	2.48E-02	1.00E-02	0.85	0.93	34.2
	A-GRU	2.45E-02	0.011	0.87	0.88	28.39
	A-LSTM	2.83E-02	0.012	0.83	0.85	33.25
	Proposed	1.60E-02	7.70E-03	0.94	0.94	13.5
Mumbai	GRU	6.65E-02	5.17E-02	0.73	0.75	34.7
	LSTM	6.65E-02	5.27E-02	0.72	0.74	35.6
	BiGRU	6.35E-02	4.64E-02	0.75	0.78	27.3
	BiLSTM	6.10E-02	4.59E-02	0.77	0.80	27.9
	VAE	5.90E-02	4.29E-02	0.78	0.80	38.3
	A-GRU	4.24E-02	0.063	0.60	0.68	37.6
	A-LSTM	7.90E-02	4.29E-02	0.59	0.68	36.7
	Proposed	5.90E-02	1.95E-02	0.79	0.81	24.69
Hyderabad	GRU	4.12E-02	3.45E-02	0.76	0.82	35.2
	LSTM	4.35E-02	3.70E-02	0.73	0.80	37.5
	BiGRU	3.75E-02	3.13E-02	0.80	0.83	30.08
	BiLSTM	3.75E-02	3.05E-02	0.81	0.83	30.10
	VAE	4.75E-02	3.80E-02	0.69	0.81	40.78
	A-GRU	4.25E-02	0.03	0.75	0.78	36.20
	A-LSTM	4.35E-02	0.035	0.73	0.75	36.8
	Proposed	3.40E-02	2.75E-02	0.85	0.85	24.85

Table 5. SO_2 pollutant based performance analysis

State	Model	RMSE	MAE	R2	EV	MAPE (%)
Chennai	GRU	2.00E-02	1.56E-02	0.95	0.95	102.4
	LSTM	2.00E-02	1.56E-02	0.95	0.95	99.5
	BiGRU	1.73E-02	1.42E-02	0.95	0.95	70.70
	BiLSTM	2.00E-02	1.49E-02	0.94	0.94	80.75
	VAE	1.70E-02	1.51E-02	0.95	0.95	101.55
	A-GRU	2.00E-02	0.0153	0.94	0.94	120.40
	A-LSTM	0.24E-02	0.0185	0.91	0.91	132.30
	Proposed	1.51E-02	1.26E-02	0.96	0.96	59.8
Delhi	GRU	2.40E-02	1.10E-02	0.75	0.76	13.5
	LSTM	2.50E-02	1.30E-02	0.66	0.78	15.60
	BiGRU	1.90E-02	8.00E-03	0.89	0.90	9.5
	BiLSTM	2.10E-02	9.00E-03	0.85	0.88	10.3
	VAE	2.40E-02	1.50E-02	0.013	0.79	30.45
	A-GRU	1.42E-02	1.24E-02	0.71	0.74	14.5
	A-LSTM	1.40E-02	1.25E-02	0.70	0.71	14.5
	Proposed	9.50E-02	7.45E-03	0.90	0.90	7.5
Mumbai	GRU	2.00E-02	1.60E-02	0.64	0.79	37.8
	LSTM	2.24E-02	1.70E-02	0.58	0.73	42.3
	BiGRU	1.73E-02	1.27E-02	0.73	0.83	26.9
	BiLSTM	2.00E-02	1.60E-02	0.62	0.80	30.8
	VAE	1.90E-02	1.60E-02	0.70	0.76	64.5
	A-GRU	2.83E-02	2.26E-02	0.30	0.65	48.8

	A-LSTM	3.00E-02	2.35E-02	0.24	0.63	50.8
	Proposed	1.25E-02	9.60E-02	0.87	0.87	27.9
Hyderabad	GRU	2.45E-02	1.73E-02	0.70	0.73	69.15
	LSTM	2.45E-02	1.96E-02	0.65	0.69	80.15
	BiGRU	2.00E-02	1.35E-02	0.78	0.80	50.07
	BiLSTM	2.00E-02	1.32E-02	0.76	0.78	49.04
	VAE	2.22E-02	1.61E-02	0.74	0.74	43.9
	A-GRU	2.45E-02	1.85E-02	0.67	0.70	75.56
	A-LSTM	2.65E-02	1.84E-02	0.65	0.66	72.9
	Proposed	1.98E-02	1.07E-02	0.78	0.80	37.89

Table 6. O_3 pollutant based performance analysis

State	Model	RMSE	MAE	R2	EV	MAPE (%)
Chennai	GRU	3.00E-02	2.40E-02	0.95	0.95	10.02
	LSTM	3.20E-02	2.57E-02	0.94	0.95	1085
	BiGRU	2.65E-02	2.08E-02	0.96	0.96	9.30
	BiLSTM	2.65E-02	2.15E-02	0.96	0.96	9.35
	VAE	2.98E-02	2.10E-02	0.95	0.95	8.5
	A-GRU	3.16E-02	0.0253	0.94	0.95	11.9
	A-LSTM	3.61E-02	2.1185	0.93	0.94	14.20
	Proposed	2.64E-02	1.26E-02	0.96	0.96	8.20
Delhi	GRU	3.20E-02	2.40E-02	0.92	0.93	9.20
	LSTM	3.20E-02	2.30E-02	0.93	0.93	9.5
	BiGRU	3.20E-02	2.10E-03	0.94	0.94	8.1
	BiLSTM	3.20E-02	2.20E-03	0.94	0.94	8.6
	VAE	3.20E-02	2.40E-02	0.92	0.93	9.2
	A-GRU	2.83E-02	0.02	0.94	0.94	9.2
	A-LSTM	3.75E-02	0.30	0.89	0.90	11.8
	Proposed	2.53E-02	1.99E-03	0.95	0.95	6.8
Mumbai	GRU	5.66E-02	4.65E-02	0.85	0.85	32.6
	LSTM	5.39E-02	4.41E-02	0.86	0.86	30.7
	BiGRU	5.57E-02	4.55E-02	0.85	0.86	30.9
	BiLSTM	5.66E-02	4.61E-02	0.85	0.85	33.14
	VAE	5.40E-02	4.40E-02	0.86	0.86	32.10
	A-GRU	5.75E-02	0.0470	0.84	0.86	30.3
	A-LSTM	5.65E-02	2.35E-02	0.85	0.87	30.0
	Proposed	5.05E-02	4.15E-02	0.88	0.88	28.85
Hyderabad	GRU	3.30E-02	2.85E-02	0.85	0.87	18.15
	LSTM	3.45E-02	2.77E-02	0.85	0.87	18.1
	BiGRU	3.16E-02	2.55E-02	0.88	0.88	15.05
	BiLSTM	3.33E-02	2.60E-02	0.87	0.88	16.3
	VAE	2.60E-02	2.10E-02	0.92	0.93	12.7
	A-GRU	3.71E-02	0.030	0.85	0.85	17.5
	A-LSTM	3.90E-02	0.0315	0.83	0.84	19.5
	Proposed	2.50E-02	2.08E-02	0.92	0.93	11.5

Table 7. Multivariate performance analysis

Timestep	Model	RMSE	MAE	R2	EV	MAPE (%)
3	GRU	14.5	6.13	0.85	0.85	25
	LSTM	15.3	6.64	0.83	0.84	27
	BiGRU	13.6	5.75	0.87	0.87	23
	BiLSTM	13.6	5.70	0.87	0.87	24
	VAE	14.7	6.50	0.85	0.85	25
	A-GRU	14.2	6.15	0.85	0.85	25
	A-LSTM	13.8	5.85	0.86	0.86	25
	Proposed	12.9	5.30	0.88	0.88	22
6	GRU	14.7	6.50	0.85	0.85	28
	LSTM	13.90	5.90	0.86	0.86	26
	BiGRU	13.69	5.75	0.86	0.86	24
	BiLSTM	13.19	5.57	0.87	0.87	23

	VAE	14.5	6.30	0.85	0.86	27
	A-GRU	13.6	5.9	0.86	0.86	25
	A-LSTM	13.8	5.9	0.86	0.86	25
	Proposed	12.5	5.18	0.88	0.89	20
9	GRU	14.8	6.5	0.85	0.85	28
	LSTM	13.5	5.8	0.87	0.87	24
	BiGRU	139	6.1	0.86	0.86	29
	BiLSTM	13.3	5.6	0.87	0.87	23
	VAE	13.5	5.5	0.88	0.88	25
	A-GRU	14.4	5.8	0.87	0.87	27
	A-LSTM	12.5	5.9	0.85	0.86	22
	Proposed	11.6	5.0	0.90	0.90	21
12	GRU	13.4	5.7	0.87	0.87	26
	LSTM	13.4	5.7	0.87	0.87	25
	BiGRU	13.3	5.7	0.87	0.87	25
	BiLSTM	13.5	5.9	0.87	0.87	27
	VAE	12.6	5.4	0.88	0.88	26
	A-GRU	13.3	5.8	0.87	0.87	28
	A-LSTM	14.2	5.9	0.85	0.86	25
	Proposed	11.9	4.8	0.90	0.90	21

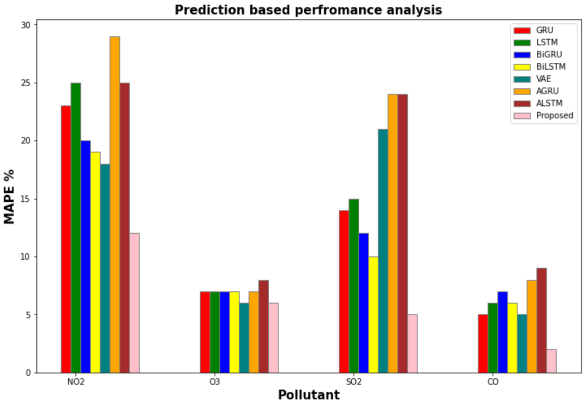


Figure 4. MAPE based performance analysis

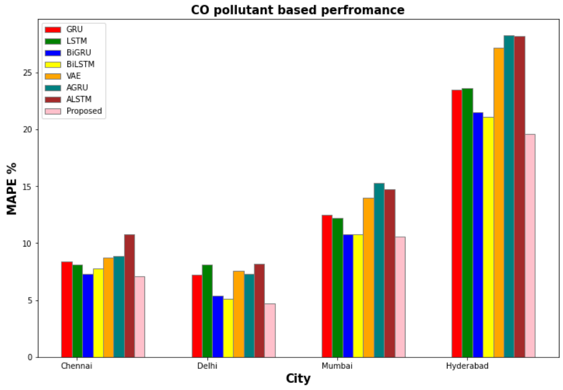


Figure 5. Mapping MAPE for CO pollutant

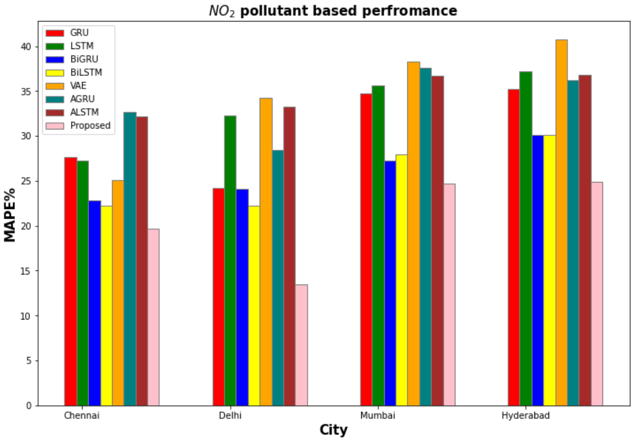


Figure 6. Mapping MAPE for NO_2 pollutant

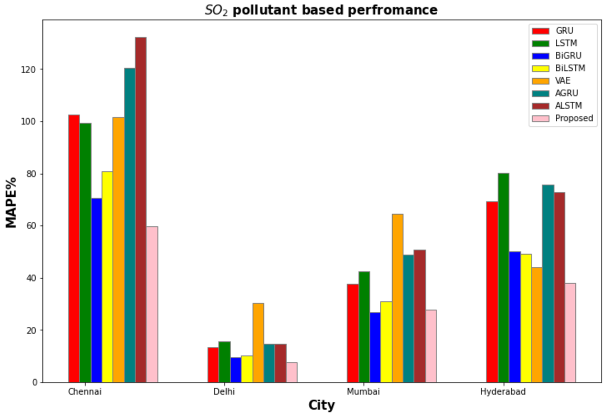


Figure 7. Mapping MAPE for SO_2 pollutant

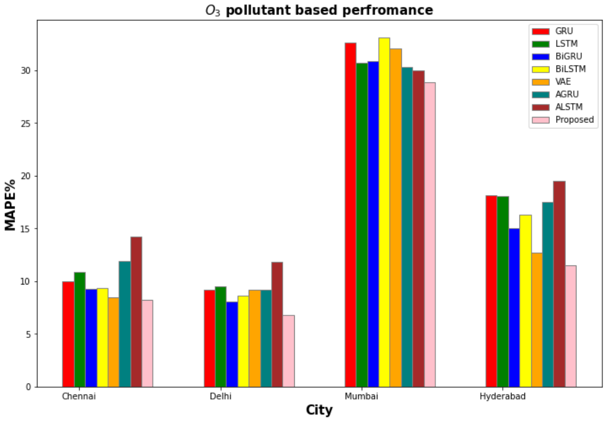


Figure 8. Mapping MAPE for O_3 pollutant

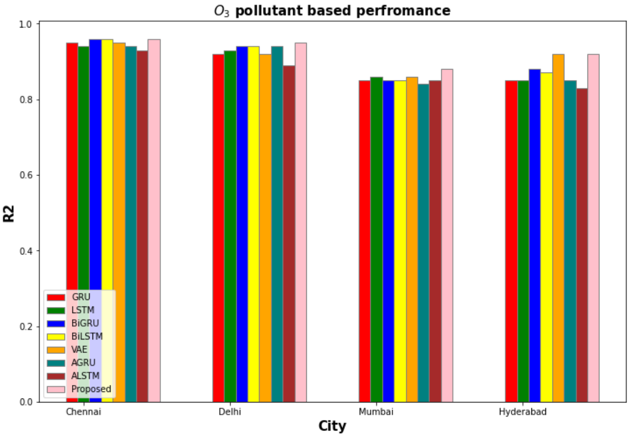


Figure 9. Mapping R_2 for SO_3 pollutant

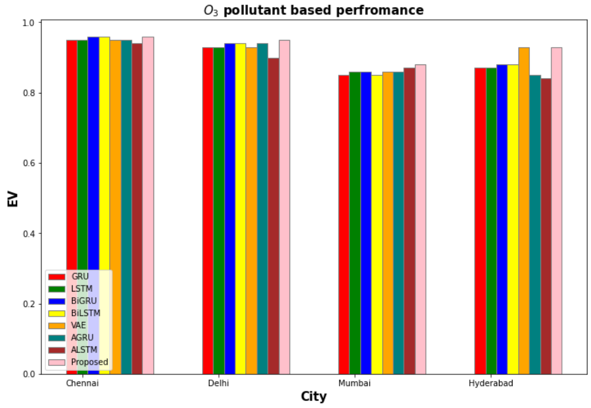


Figure 10. Mapping EV for SO_2 pollutant

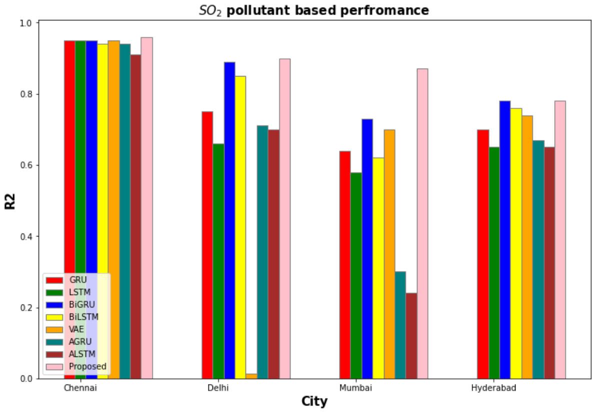


Figure 11. Mapping R_2 for SO_2 pollutant

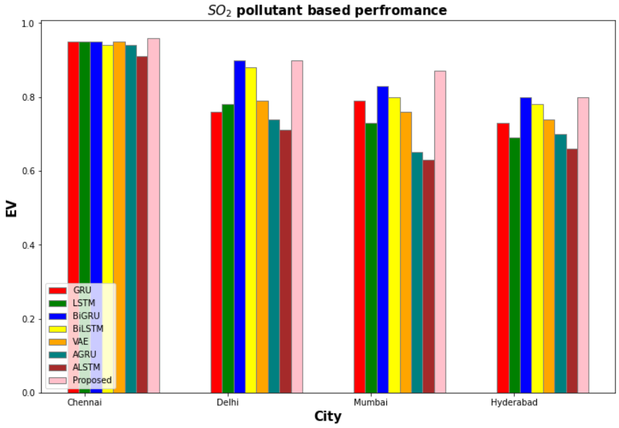


Figure 12. Mapping *EV* for SO_2 pollutant

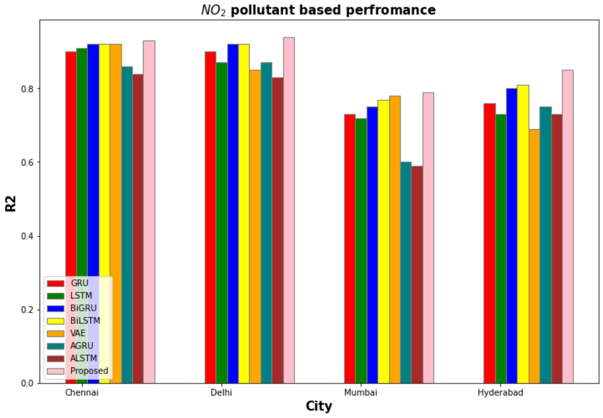


Figure 13. Mapping R_2 for NO_2 pollutant

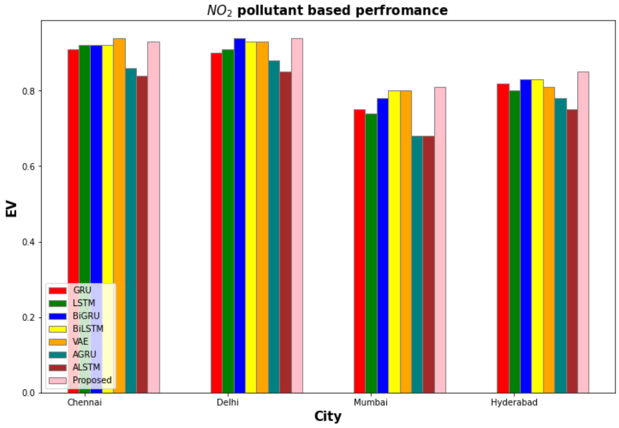


Figure 14. Mapping *EV* for NO_2 pollutant

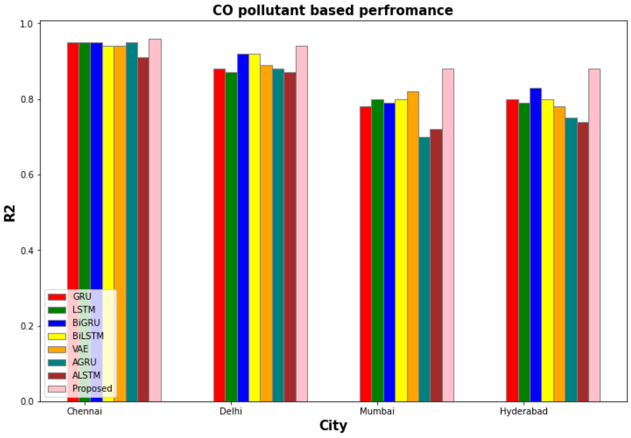


Figure 15. Mapping R_2 for CO pollutant

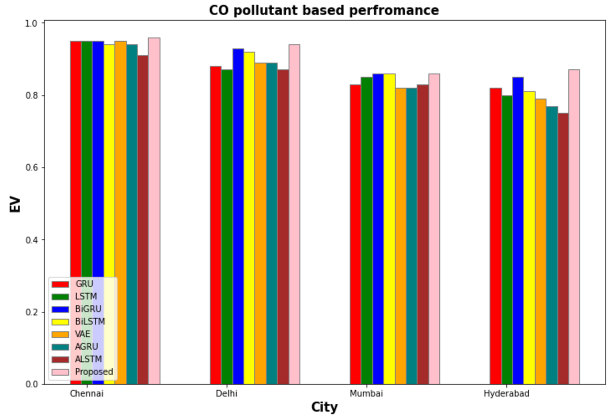


Figure 16 Mapping EV for CO pollutant

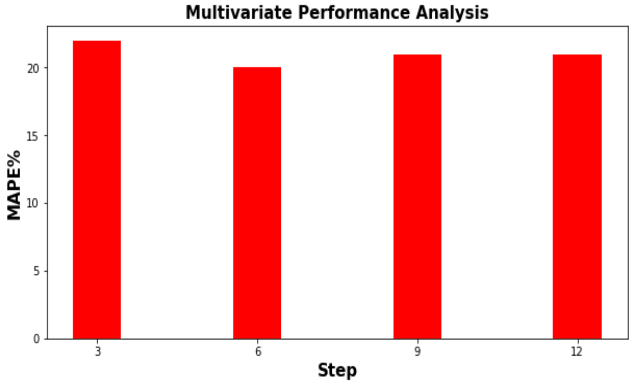


Figure 17 Multivariate performance analysis

Tables 3 to 7 depicts the performance comparison of the anticipated model based on pollutant measurements. The model shows that the percentage of NO_2 variability is superior to EV and R2 while other metrics like recorded as RMSE= 8.5, MEA = 6.2, MBE = -3.2, and RMBE = -3.1. However, it is observed that the data processing carried out using BiGRU and BiLSTM model is superior time-dependencies while predicting NO2 (i.e. EV = 95.6% and R2 = 95%) compared to other unidirectional RNN such as GRU and LSTM (EV = 92% and R2 = 90%). Also, some conclusions are attained for predicting the O_3 concentration level, which shows 96% variability followed by 95% BiLSTM while the other score is lesser than 93%. Also, MPAE of the anticipated model is recorded as 6%, while the MPAE is recorded as 7%, where the proposed model shows higher performance. It is observed that the performance of the anticipated model is recorded for CO concentration levels. Subsequently, the validation of SO_2 predicts values from the expected model specify the superior prediction quality by attaining R2 → 0.97 and reduced error rate (MAE = 0.45, RMSE = 0.55 and MAPE = 5%). Similarly, the bidirectional (GRU and LSTM) models perform superior to the unidirectional model by attaining MPAE of BiGRU is 12% and BiLSTM is 10%, respectively (See Figure 4 to Figure 16).

The successive experimentation is provided for a comparative evaluation for predicting the pollutants' concentration levels from the multiple inputs. The outcomes confirm the most satisfactory performance of the anticipated model compared to specific traditional approaches devoid of accessible attention model and other standard learning models by attaining higher EV and R1 and lower mean error of provided experimentations. Also, it is more interesting to sense superior performance from the proposed bi-directional attention model than the unidirectional model (GRU and LSTM). The outcomes confirm that the proposed model outperforms other approaches like GRU and LSTM models. The results testify that the performance of the anticipated model is superior to different methods. The final experimentation evaluates the expected model's potential in predicting pollutants simultaneously using historical pollutant data from all the pollutants. This work considers the online dataset for pollutants and omits other pollutants as it provides relatively the same outcomes. The foremost advantage of the multivariate pollution prediction is using only the multivariate prediction model while predicting various pollutants simultaneously evaluated to different univariate forecasts need a model for the time-series data (See Figure 17). Moreover, multivariate prediction is relatively challenging to correlate between multiple variables' time and variable dependencies. Here, an essential variable can influence the prediction accuracy, i.e. amount of prediction data utilized to identify the previous pollutant values. Here, the prediction performance is evaluated for various timestamp values. It is observed that the proposed model records the superior score with the highest EV and R2 for all experimentation and shows better performance with 12 timestamps for multivariate prediction of all pollutants.

Moreover, the proposed model with attention-based BiLSTM is evaluated in this investigation. As observed, the conventional approach devoid of an attention mechanism consumes less time than other approaches with the attention model due to the computational cost associated with the attention model. Specifically, while concluding the experimentation using the implemented PC, i.e. average execution time is 0.0050. Therefore, the average time consumed by the attention model is 15ms. Subsequently, conducting the experimentation with the well-equipped GPU has an average execution time lesser than 10^{-5} seconds. While performing time-series analysis, the attention model is not time-consuming. Finally, overall prediction outcomes depict the superior ability thatrelies on inferences towards data probability distribution of the provided pollutant time-series with the proposed mechanism integrated at a higher level to emphasize and highlight the feature correlation among the data points of the provided sequence. It is traced that the proposed model with attention mechanism enhances the time-dependencies modelling devoid of any memory cell connections. It needs to be highlighted that the superior prediction performance of the anticipated model. When the correlations among the pollutants are higher, the performance is higher. Else, it is lower. It may be enhanced by considering multiple variables. The proposed model is utilized to predict and data is processed in a particular direction. A Bidirectional Long Short-Term Memory (Bi-LSTM) network's time and computational complexity can be expressed in terms of the number of hidden units (h) and sequence length (n). Following are the main complexities:

Time Complexity: O(n·h2) is a common way to define the time complexity of processing a single input sequence of length n. This is due to the fact that the complexity of the forward and backward passes of the Bi-LSTM is quadratic in the number of hidden units and involves matrix multiplications of the input with weight matrices in each time step.

Computational Complexity: The sum of the forward and backward passes has computational complexity O(n). This covers the processes of updating the cell state and calculating the input, output, and forget gates. . In the proposed work, these complexities has been reduced with the help of distributed processing mechanism.

5. Conclusion

Air quality needs to be predicted because it is difficult due to unpredictability, dynamic environment, and time and space variability of pollutants. Air pollution shows consequences on animals, humans, monuments, plants, the environment, and the climate, which needs to be monitored and analyzedwith the consistent quality of air, particularly in the countries in developing states. Moreover, in India, there are only fewer researchers who gain attracted to authors to predict and analyze the air quality. Currently, the 23 Indian cities provide the data related to air pollution. First, the dataset needs to be clean and pre-processed to fill the values of NAN, the outliers are addressed, and the data values need to be normalized. Thus, the feature extraction approach is processed for filtering the AQI, which affects the pollutants. The exploratory data analyses are used to identify the different patterns available in the dataset.

The dataset split to train-test subsets is done using the respective ratio of 75% to 25%. The AQI prediction is based on a deep learning approach, and the analysis with comparison is provided. The learning model gives training and testing subsets outcomes concerning the standard measures. The traditional statistic error measures called, RMSE, MAE, R2, and EV, are determined for accessing and comparing the models' performances. The best performer is considered the dense model by obtaining the optimal values in both the testing and training stages. The proposed model performs well in a relative way during the training phase. In this phase, the proposed model obtains better outcomes in the target prediction for MAPE, R2 and EV. The current study aims to contribute by mentioning the air quality analysis and the prediction in India (four states) to the literature that is not researched properly. The deep learning approaches are used by extending this work to predict the air quality because there is only less work that concentrates on multivariate data. However, this work pretends to fulfil the research gap identified in the existing approaches. The primary research constraint is the selection of states for air pollutant prediction, and some other metrics like accuracy and statistical measures need to be evaluated. Thereby, the major limitation of the proposed work is, determining the optimal hyperparameters for both the convolutional and LSTM layers can be challenging. This includes setting the number of layers, growth rate in DenseNet, number of hidden units in Bi-LSTM, learning rates, and dropout rates. This will be enhanced during the futuristic research.

References

Asokan R. and Preethi P. (2021). Deep learning with conceptual view in meta data for content categorization. In Deep Learning Applications and Intelligent Decision Making in Engineering (pp. 176–191). IGI Global.

Biancofore F, Busilacchio M, Verdecchia M *et al.* (2017). Recursive neural network model for analysis and forecast of PM10 and PM2.5. Atmospheric Pollution Research, **8**, 652–659. https://doi.org/10.1016/j. apr.2016.12.014

Department of Economic and Social Affairs: Urban Population Change; 2018. https://www.un.org/development/desa/en/news/population/2018-revision-of-world-urbanization-prospects.html. Accessed 20 Oct 2021.

F. Harrow, Y. Sun, A. S. Hering, M. Madakyaru *et al.* (2020). Statistical process monitoring using advanced data-driven and deep learning approaches: theory and practical applications. Elsevier.

Gollakota A.R.; Gautam S. and Shu C.-M. (2020). Inconsistencies of e-waste management in developing nations–Facts and plausible solutions. J. Environ. Manag, **261**, 110234

Hernandez A. Mendez R. Zalakeviciute. and A. M. Diaz-Marquez. (2020). "Analysis of the information obtained from PM 2.5 concentration measurements in an urban park," IEEE Transactions on Instrumentation and Measurement, vol. 69, no. 9, pp. 6296–6311.

Huang C-J, Kuo P-H. (2018). A deep CNN-LSTM model for particulate matter (PM2.5) forecasting in smart cities, Sensors, **18**, 2220. https://doi.org/10.3390/s18072220

Kok I, Simsek MU. and Ozdemir S. (2017). A deep learning model for air quality prediction in smart cities. In: 2017 IEEE International Conference on Big Data (Big Data). IEEE, pp 1983–1990.

Kulkarni G.E., Muley A.A., Deshmukh N.K., Bhalchandra P.U. (2018). Autoregressive integrated moving average time series model for forecasting air pollution in Nanded city, Maharashtra India. Model, Earth Syst. Environm, **4(4)**, 1435–1444. https://doi. org/10.1007/s40808-018-0493-2.

Kulurkar P., kumar Dixit C., Bharathi V. C., Monikavishnuvarthini A., Dhakne A. and Preethi P. (2023). AI based elderly fall prediction system using wearable sensors: A smart home-care technology with IOT. Measurement: Sensors, 25, 100614.

Li T, Hua M. and Wu X. (2020). A Hybrid CNN-LSTM model for forecasting particulate matter (PM2.5). IEEE Access 8:26933–26940. https://doi.org/10.1109/ACCESS.2020.2971348

Li X, Peng L, Yao X *et al.* (2017). Long short-term memory neural network for air pollutant concentration predictions: method development and evaluation. Environment Pollution, **231**, 997–1004. https://doi. org/10.1016/j.envpol.2017.08.114

Liang X, Zou T, Guo B *et al.* (2015). Assessing Beijing's PM 2.5 pollution: severity, weather impact, APEC and winter heating. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, **471**, 20150257. https://doi. org/10.1098/rspa.2015.0257

Lionetto M.G.; Guascito M.R.; Caricato R.; Giordano M.E.; Bartolomeo A.R.D.; Romano MP; Conte M.; Dino A. and Contini D. (2019). Correlation of Oxidative Potential with

Ecotoxicological and Cytotoxicological Potential of PM10 at an Urban Background Site in Italy. Atmosphere, **10**, 733

Luo Z, Huang J, Hu K, *et al.* (2019). AccuAir. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, New York, pp 1842–1850

Moursi AS, Shouman M, Hemdan EE. and El-Fishawy N. (2019). PM2.5 concentration prediction for air pollution using machine learning algorithms, Menoufa J Electron Eng Res, **28**, 349–354. https://doi.org/10.21608/mjeer.2019.67375

Palanisamy P., Padmanabhan A., Ramasamy A. and Subramaniam S. (2023). Remote Patient Activity Monitoring System by Integrating IoT Sensors and Artificial Intelligence Techniques. Sensors, **23(13)**, 5869.

Preethi P. and Asokan R. (2020). Neural network oriented roni prediction for embedding process with hex code encryption in dicom images. In Proceedings of the 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, India (pp. 18–19).

Punarselvam E., Devi T. K., Britto A., Prakash N. B. and Suresh P. (2020). Segmentation Analysis Techniques and Identifying Stress Ratio of Human Lumbar Spine Using ANSYS. Journal of Medical Imaging and Health Informatics, 10(10), 2308–2315.

Punarselvam E., Sikkandar M. Y., Bakouri M., Prakash N. B., Jayasankar T. and Sudhakar S. (2023). Retraction Note to: Different loading condition and angle measurement of human lumbar spine MRI image using ANSYS.

Punarselvam E., Sikkandar M. Y., Bakouri M., Prakash N. B., Jayasankar T. and Sudhakar S. (2021). Different loading condition and angle measurement of human lumbar spine MRI image using ANSYS. Journal of Ambient Intelligence and Humanized Computing, **12**, 4991–5004.

Rodríguez-Urrego D, Rodríguez-Urrego L. (2020). Air quality during the covid-19: Pm2.5 analysis in the 50 most polluted capital cities in the world. Environ Pollut.

Roser M. (2023). Data review: how many people die from air pollution?. Our World in Data.

Ventura F. de Oliveira Pinto L. M. Soares A. S. Luna. and A. Gioda. (2019). "Forecast of daily PM 2.5 concentrations applying artificial neural networks and Holt–Winters models," Air Quality, Atmosphere & Health, vol. 12, no. 3, pp. 317–325.

Yang G, Lee H. and Lee G. (2020). A hybrid deep learning model to forecast particulate matter concentration levels in Seoul, South Korea. Atmosphere, **11**, 348. https://doi.org/10.3390/atmos11040 348