Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 4, No. 1, April : 2023

AIR POLLUTION PREDICTION USING MULTIVARIATE LSTM DEEP LEARNING MODEL

G.Naresh Research Scholar, Department of Informatics, University College of Engineering, Osmania University, Hyderabad, Telangana
Dr. B.Indira Associate Professor, Department of MCA, Chaitanya Bharathi Institute of Technology(A), Osman Sagar Rd, Kokapet, Gandipet, Telangana.
[1]nani.naresh2126@gmail.com, [2]bindira_mca@cbit.ac.in

Abstract
Air pollution prediction is the process of using data analysis and modelling techniques to forecast the level of pollutants in the air at a future time or location. Air pollution prediction using deep learning is an active area of research and has many practical applications, including improving public health, reducing environmental damage, and supporting decision-making processes for urban planning and transportation management. This paper presents a Long Short-Term Memory (LSTM) based air pollution prediction model. LSTM is a type of recurrent neural network (RNN) that can be used to predict air pollution levels. LSTM models are particularly useful for predicting time series data, such as air pollution levels measured at specific time intervals. LSTM models can be used to predict air pollution levels by learning complex patterns in the historical data and identifying the factors that contribute to high levels of pollution.

Keywords: air pollution prediction, Long Short-Term Memory (LSTM), recurrent neural network (RNN)

1. Introduction

The Urban Population website estimates that 56.15 percent of the population will be living in cities in the year 2020. The United Nations estimates that by the year 2050, cities will be home to 68% of the total population of the globe. This change in population would lead to several problems in terms of health, transportation, and the quality of the air. Air pollution is a major contributor to a wide range of adverse health effects, including difficulties breathing, early mortality, and hospitalization for heart and lung disorders. People are impacted by air pollution, but plants are far more so due to the fact that prolonged exposure to pollutants may cause damage to plant leaves [1]. The majority of the contaminants come from stationary sources of the primary air pollutants, which are dust and particles of Particulate Matter with a diameter of fewer than 10μm (PM10) and the PM2.5, which are the most dangerous because their diameter is less than PM2.5 microns, which is produced from unburned fuel and process byproducts, as well as sulphur dioxide (SO2) that is produced by these sources. Another primary air pollutant is sulphur dioxide (SO2), which is produced by these sources [2]. Nitrogen oxides (NOx), carbon monoxide (CO), and ozone are produced as byproducts of the burning of fuel. Nitrogen oxides are formed when oxygen and nitrogen combine with extreme heat (O3).

As the country with the fastest-growing industrial sector, India is responsible for a record-breaking quantity of pollution, including carbon dioxide (CO2), PM2.5, and other hazardous air pollutants. According to the Indian air quality standard, pollutants are indexed in terms of their scale. These Air Quality Indexes (AQI) indicate the levels of major pollutants that are present in the atmosphere. The air quality of a particular state or country is a measure of the effect that pollutants have on the respective regions [3]. There are many different gases in the atmosphere that contribute to the pollution of our environment. Every kind of pollution has its own unique index and scale, with varying degrees of severity. The main pollutants' AQI indices are obtained; with each individual AQI, the data may be classified in accordance with the limitations.

For human health, an efficient system for tracking and calcification of air pollution is crucial. However, the mechanism and process of PM2.5 formation are extremely complicated due to the complexity of its properties. These properties, such as non-linear properties in time and space, have a significant impact on the accuracy of prediction and make the mechanism and process extremely difficult to understand. At this time, the majority of data gathering on air quality is done at micro-stations [4]. However, such in-situ monitoring is less viable in the bulk of places of concern as a result of the high material and set-up costs of modern sensors. This constitutes a considerable financial burden for poor and growing countries over the long run. It is feasible to employ image-based systems for monitoring air quality as a backup when gauges are either unavailable or not performing adequately. This is something that is doable. In recent times, there have been several efforts made to develop monitoring technology at cheap cost that is specific to air pollution [5].

Deep neural networks, particularly Convolutional Neural Network (CNN), which have strong data processing capabilities, have been more used in image classification and identification as machine learning has progressed. The CNN has been used extensively in research in the domains of computer vision and image processing due to its credible performance in tackling a variety of interesting tasks on classification and estimate. In recent years, there has been a rise in interest in using machine learning [6,7] and deep learning [8] techniques for the purpose of monitoring air quality. Image processing has been used in a great number of studies to categories or estimate levels of air pollution. In addition, an image-based air pollution estimate offers a positive outlook for the future; yet, very few research of this kind have been carried out within this environment. Because of this, there is a pressing need for more research into image-based estimations of air quality in order to improve their accuracy and dependability. Recently, numerous automated methods have been proposed as effective solutions to handle the issues associated with crack identification in practice. This is mostly attributable to the fast expansion of deep learning algorithms and the advancements in computer vision technology.

2. Literature

Air pollution forecasting using deep learning has been an active area of research in recent years. Deep learning models are well suited for air pollution forecasting because they are able to capture long-term dependencies in time series data. One of the key advantages of using deep learning for air pollution forecasting is their ability to handle missing data. Air pollution data can be incomplete or noisy, and traditional time series models may struggle to handle this type of data. Deep learning models, however, have been shown to be effective at handling missing data and are able to make accurate predictions even when the input data is incomplete.

An Aggregated LSTM model (ALSTM), based on the LSTM deep learning technique, was proposed by Yue-Shan Chang [9]. In this novel model, authors incorporate the stations for external pollution sources, surrounding industrial zones, and local air quality monitoring stations. For early forecasts based on outside sources of pollution and data from adjacent industrial air quality monitors, authors use three LSTM models to increase prediction accuracy. The authors tested our novel ALSTM model in comparison to SVR (Support Vector Machine based Regression), GBTR (Gradient Boosted Tree Regression), LSTM, and others in the prediction of PM2.5 over 1–8 hours, and they assessed them using a variety of evaluation methods, including MAE, RMSE, and MAPE.

Recurrent neural networks (RNNs) with long short-term memory units are used by Tien-Cuong Bui [10] as a framework for using knowledge from time-series data of air pollution and meteorological data in Daegu, Seoul, Beijing, and Shenyang. Additionally, a key component of our prediction engine is the encoder-decoder paradigm, which is comparable to machine understanding issues. The accuracy of different configurations' predictions is finally looked at by the writers. When predicting a large number of timesteps in the future, the trials prohibit the effectiveness of integrating many layers of RNN on prediction models. This study serves as a strong impetus to continue studying urban air quality and to assist the government in using that knowledge to implement sensible policy.

Jingyang Wang [11] has presented a new CT-LSTM technique in which LSTM network model and the chi-square test (CT) are combined to construct the prediction model. CT is used to identify the variables that affect air quality. The LSTM network model is trained using the hourly air quality data and meteorological data from the year 2017 to 2018. The LSTM network model is assessed using data in the year 2019. From January 1 to December 31, 2019, the AQI level in Shijiazhuang, Hebei Province, China, is forecasted using five different techniques (SVR, MLP, BP neural network, Simple RNN, and the novel technique presented in this study). The five prediction findings are then contrastively analyzed.

Duen-Ren Liu [12], proposes a new wind-sensitive attention mechanism using LSTM neural network model to forecast the air pollution - PM2.5 concentrations by taking into account the impact of wind direction and speed on the variations of spatial-temporal PM2.5 concentrations in nearby locations. An LSTM neural network then generates preliminary predictions for PM2.5 based on nearby pollution, which are then "paid attention" to. Finally, we use an ensemble learning technique based on eXtreme Gradient Boosting (XGBoost) to combine the preliminary predictions with weather forecasting to generate second phase predictions for PM2.5.

In order to track and collect real-time data on air pollution concentrations from diverse locations and to utilize this information to predict future air pollutant concentrations, Sagar VBelavadi [13] presents a scalable architecture. To get information on air quality, two sources are employed. The first is a wireless sensor network with sensor nodes placed around Bengaluru, a city in South India, that collects and transmits pollution concentrations to a server. The second source is the Government of India's Open Data project, which includes the collection and dissemination of real-time data on air quality. Hourly average concentrations of several air contaminants are provided by both sources. A LSTM-RNN model was selected to carry out the job of air quality forecasting because to its shown track record of performance with time-series data. The model's performance in two areas with very different temporal fluctuations in air quality is rigorously examined in this research.

LSTM and deep autoencoder (DAE) techniques were used by Thanongsak Xayasouk [14] to create models to predict fine PM concentrations, and the model outputs were compared in terms of root mean square error (RMSE). The models were applied to hourly air quality data collected from 25 stations in Seoul, South Korea, between January 1, 2015, and December 31, 2018, by the authors. At an ideal learning rate of 0.01 for 100 epochs with batch sizes of 32 for the LSTM model and 64 for the DAEs model, fine PM concentrations were forecasted for the ten days that followed this period. The proposed models accurately predicted the concentrations of fine PM, with the LSTM model performing somewhat better.

3. Proposed Model

The LSTM networks are very effective at representing sequential data. LSTMs use a unique kind of memory cell known as an LSTM cell to record long-term dependencies in sequential data. Information entering and leaving these cells is managed by three gates: input, forget, and output gates. The LSTM may recall or forget prior knowledge as required thanks to the gates, which are controlled by learning weights and have the ability to selectively allow or restrict the flow of information. The capacity of LSTMs to accommodate missing or noisy data is one of its key features. In the case of missing or noisy data, traditional RNNs, such as Elman networks, may find it difficult to sustain long-term relationships. To sustain long-term dependencies despite absent or noisy input, LSTMs have the capacity to selectively recall or forget information as necessary.

LSTMs have recently been enhanced using an attention mechanism and are now known as Attention-LSTM. To selectively concentrate on significant characteristics in the input data, the Attention-LSTM model makes use of an attention mechanism. As a result, the model can anticipate outcomes with more accuracy. Overall, LSTM networks have been shown to be useful in a variety of tasks and are a strong tool for modelling sequential data. They can manage noisy and missing data and can identify long-term relationships in the data. But in order to train properly, LSTMs may need a lot of data and be computationally costly.

3.1 LSTM Architecture

The LSTM units are a kind of building unit that can be used in RNN layers. An LSTM network is a generic term for an RNN that is constructed using LSTM units. The LSTM neural network differs from more conventional RNN neural networks in that each neuron in the LSTM network functions as a memory cell. The LSTM connects the neurons that are active now to the data and information from before. Input gate, forget gate, and output gate are the three gates that are contained inside each neuron. The issue of the data becoming dependent over the long term may be solved by the LSTM by making use of its internal gate. Next, we will discuss the LSTM's internal gates and explain how the LSTM design may be used to address issues with long-term dependencies. The basic LSTM architecture is depicted in Figure 1.

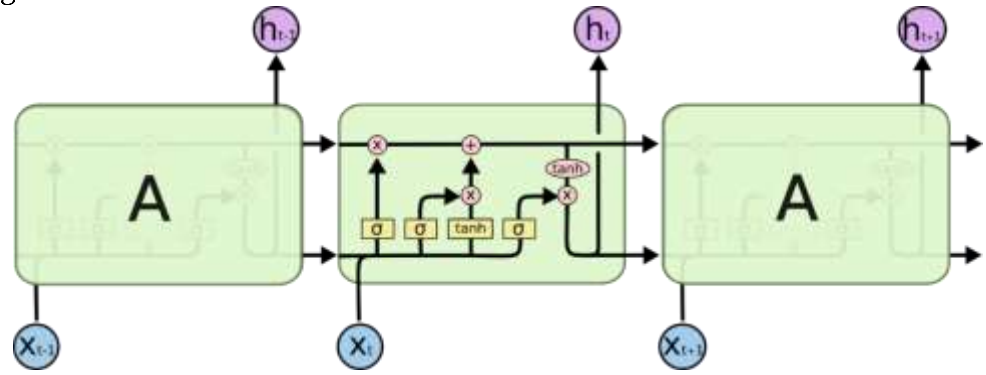


Figure 1: Basic LSTM Architecture

(a) Input gate: In an LSTM model, the input gate controls the flow of information into the memory cell. It is a sigmoid function that takes as input the current input and the previous hidden state. The output of the sigmoid function is a value between 0 and 1, which represents the amount of information that will be allowed to flow into the memory cell. When the input gate is close to 0, it means that little or no information is allowed to flow into the memory cell, and the previous state of the cell is maintained. When the input gate is close to 1, it means that a large amount of information is allowed to flow into the memory cell, and the previous state of the cell is overwritten.

(b) Forget gate: In an LSTM model, the forget gate controls the flow of information out of the memory cell. When the forget gate is close to 1, it means that a large amount of information is allowed to flow out of the memory cell, effectively forgetting the previous state of the cell. When the forget gate is close to 0, it means that little or no information is allowed to flow out of the memory cell, effectively remembering the previous state of the cell. This gate is responsible for determining what data should be saved or is significant, as well as what data the network should forget about or discard. One of many activation functions, such as a sigmoid function, a ReLu function, or a $\tan h$ function, is used to select which data will be kept.

(c) Output gate: There is a limit to the amount of data that can be produced from an LSTM system. This gate determines which output from the unit is suitable and sends that information on to the next unit. In the beginning, the values of the previous concealed state as well as the current state are input into the third sigmoid function. The tanh function is then applied to the newly created cell state that was derived from the existing cell state. Multiplication is performed point-by-point on both of these outputs. The network makes its decision on the information that the hidden state should carry on the basis of the final value. The ability to forecast outcomes relies on this hidden condition. At last, the newly discovered cell state as well as the newly discovered hidden state are passed forward to the subsequent time step.

3.2 Proposed LSTM Architecture

In the process of backpropagation, the vanishing gradient issue is the one that LSTM is mainly designed to address. A gating mechanism is used by LSTMs in order to regulate the memorization process. Through the use of gates that may open and shut, information can be read, written, and stored in LSTMs. Memory is stored in an analogue manner by these gates, which also provide element-wise multiplication using sigmoid ranges between 0 and 1. Because of its inherently differentiable character, analogue is a fine fit for backpropagation. In this work, the Multivariate LSTM model is used to forecast the air pollution.

Multivariate LSTM (MV-LSTM) is a type of LSTM that is specifically designed to handle multiple input variables, each of which may have a different set of dependencies and patterns in the data. In contrast, a traditional univariate LSTM is only designed to handle a single input variable. In a MV-LSTM, each input variable is processed by its own LSTM cell, and the outputs of all the cells are concatenated and processed by another LSTM cell. This allows the MV-LSTM to capture the dependencies and patterns in each of the input variables, and to use that information to make predictions about the future values of all the variables.

MV-LSTM is particularly useful for tasks that involve time series forecasting with multiple variables, such as stock market predictions or weather forecasting. In these tasks, the relationships between the variables are often complex and interdependent, and a MV-LSTM can capture these relationships and use them to make accurate predictions. Overall, MV-LSTM provides a more powerful and flexible way of handling multiple input variables compared to traditional univariate LSTMs, and can be used to achieve higher accuracy in time series forecasting and other tasks that involve multiple input variables.

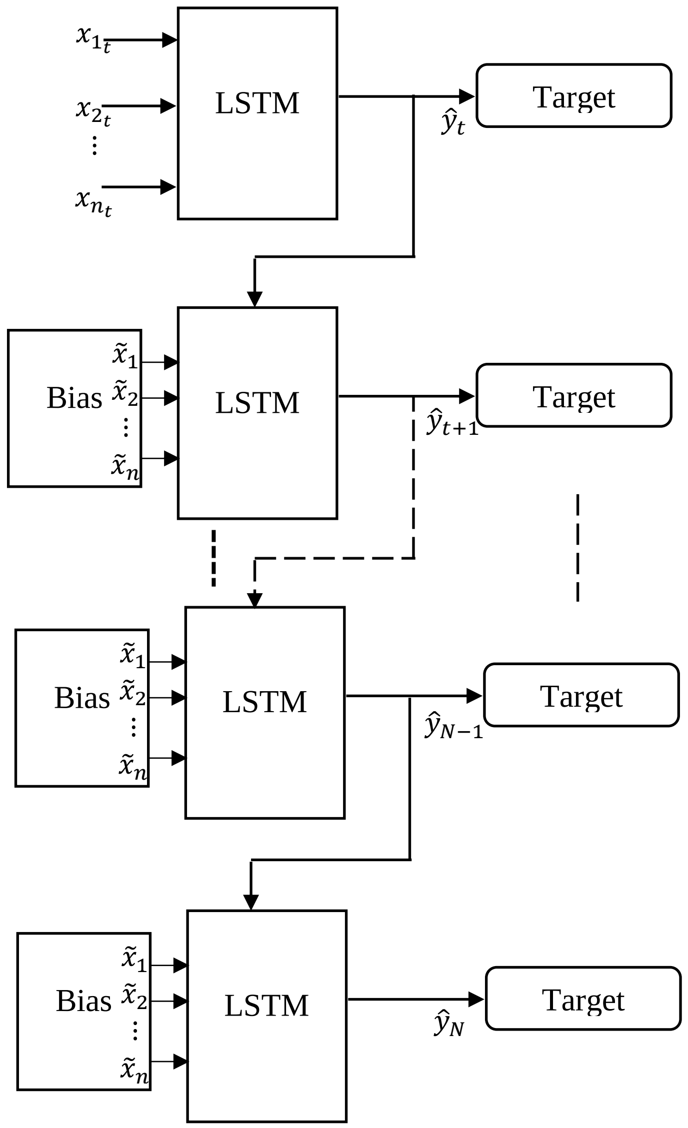


Figure 2: Proposed MV-LSTM model

The description of the proposed multivariate LSTM recurrent neural networks may be seen in Figure 2. The observed predictor characteristics are used as an input for the first LSTM; however, the expectation bias term $e_{i,t}$ at the current time together with the value of the output from the prior LSTM are used as inputs for all subsequent LSTMs. Here, we will refer to this new concept as $\tilde{x}_{i,t}$:

$$\tilde{x}_{i,t} = \begin{cases} x_{i,t} \ at \ \ t = 0 \\ e_{i,t} \ at \ \ t \neq 0 \end{cases} \tag{1}$$

Where $e_{i,t}$ is the result of applying the expectation bias function to the feature I at the given time t. The following is the generated model:

$$\hat{y}_{t+1} = LSTM\left(\hat{x}_{1,t}, \hat{x}_{2,t}, \ldots\ldots, \hat{x}_{n,t}, \hat{y}_{,t}\right) \tag{2}$$

Where n is the total number of features and $\hat{y}$ denotes the value that has been predicted.

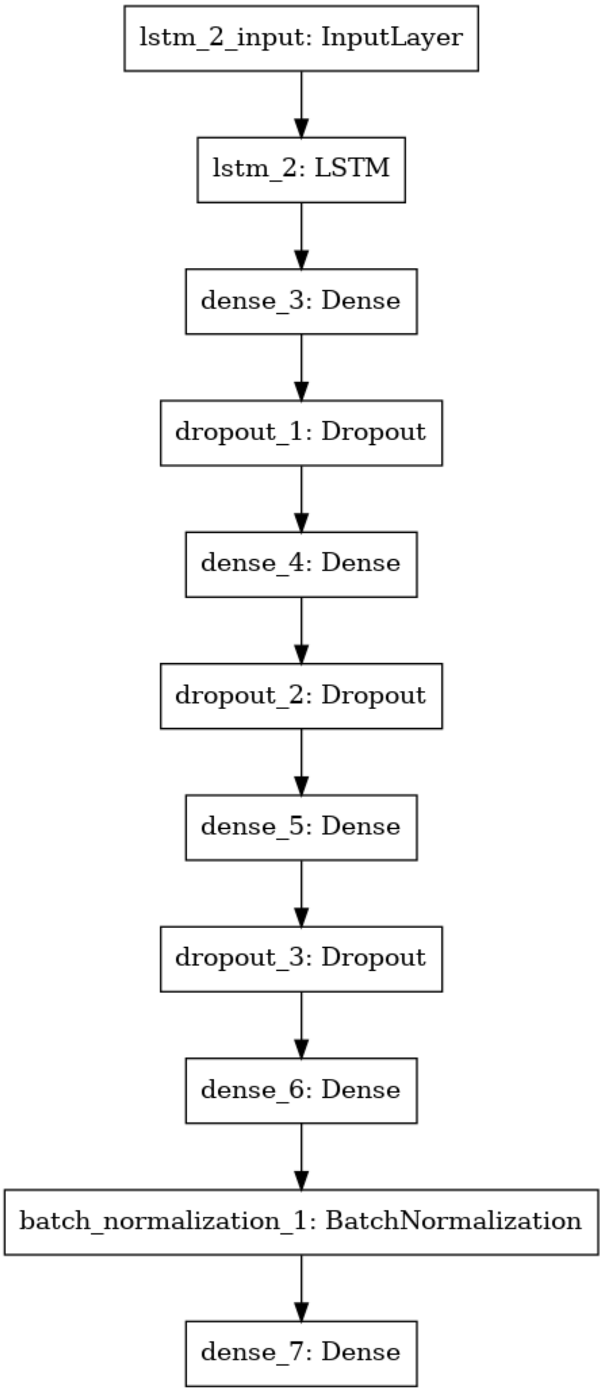


Figure 3: Proposed multi variate LSTM model

The goal of optimization is to reduce as much as possible the disparity between the output that was predicted and the output that was originally delivered.

$$minimize\ \left(loss(\hat{y}, y)\right) \tag{3}$$

The layers in the proposed model are able to remember information from previous inputs and use it to inform the processing of future inputs. This ability to retain long-term dependencies in the data makes LSTMs particularly useful in tasks such as language translation, speech recognition, and sentiment analysis. Dense layers are fully connected layers where each neuron is connected to every neuron in the previous layer. They are commonly used in neural networks for classification and regression tasks, where the input data is high-dimensional and the output is a scalar or vector. Dense layers allow the network to learn complex, non-linear relationships between the input and output data.

Dropout layers are used to prevent overfitting in neural networks. During training, a certain percentage of the neurons in the layer are randomly deactivated, which forces the remaining neurons to learn more robust features that are not dependent on any single neuron. This helps to prevent the network from memorizing the training data and enables it to generalize better to new, unseen data. Overall, these three types of layers can work together to enhance the performance of a neural network. LSTM layers can handle sequential data and retain long-term dependencies, dense layers can learn complex, non-linear relationships between the input and output data, and dropout layers can prevent overfitting and improve generalization.

4. Simulation Results

The training model and data processing for the proposed MV-LSTM model are described in this section. The basic objective of a network's training process is to reduce loss, either in terms of error or cost, that is shown in the output when training data is transmitted through it. We determine the gradient, or loss relative to a certain set of weights, then modify the weights appropriately. We continue this process until we find the best set of weights, for which the loss is minimal.

The gradient may sometimes be nearly nonexistent. It should be noted that a layer's gradient relies on certain elements in the preceding layers. The outcome, or gradient, will be considerably less if any of these components are tiny (less than 1). The scaling effect is what's recognized for this. A lower number is obtained when this gradient is multiplied by the learning rate, which is in and of itself a negligible value between 0.1 and 0.001. As a result, the change in weights is rather minor and produces about the same output as previously. The weights are adjusted to a value above the ideal value if the gradients have values that are relatively big owing to the high component values. This is referred to as the "exploding gradients issue." The neural network unit was rebuilt such that the scaling factor was maintained at one in order to prevent this scaling impact.

4.1 Dataset

The dataset used in this work provides an example of a data set for meteorological conditions, which includes columns and characteristics such as pollution, temperature, wind speed, precipitation (snow and rain), and dewpoint. Now that we have the data, we will use a method called multivariate LSTM time series forecasting to determine how much pollution will be in the air over the next several hours, taking into account factors such as temperature, humidity, wind speed, precipitation types, and snowfall. The data sample format is reported in Table 1. The data sample visualization graphs are depicted in Figure 3.

Table 1: Dataset sample format

date	pollution	dew	temp	press	wnd_dir	wnd_spd	snow	rain	
0	02-01-2010 00:00	129	-16	-4	1020	SE	1.79	0	0
1	02-01-2010 01:00	148	-15	-4	1020	SE	2.68	0	0
2	02-01-2010 02:00	159	-11	-5	1021	SE	3.57	0	0
3	02-01-2010 03:00	181	-7	-5	1022	SE	5.36	1	0
4	02-01-2010 04:00	138	-7	-5	1022	SE	6.25	2	0

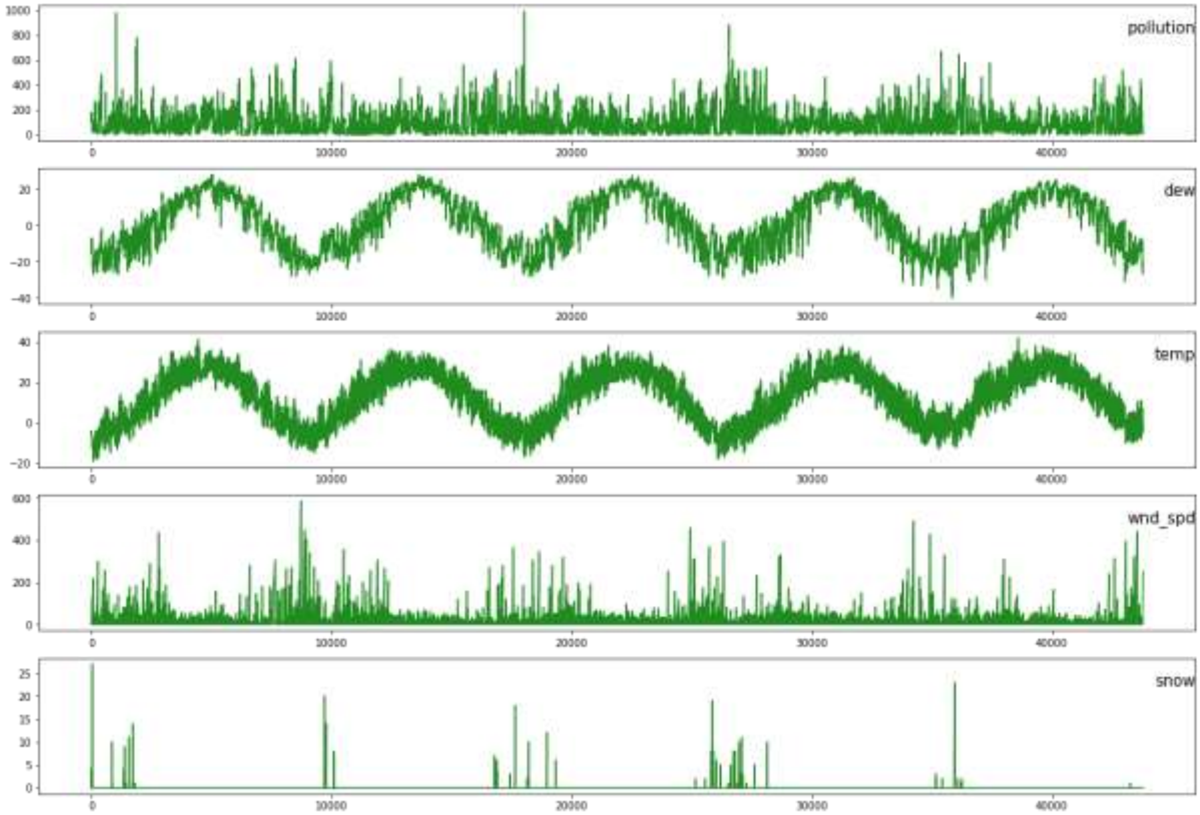


Figure 3: Data samples visualization graphs

A graph that displays the increase and fall of various pollutants' concentrations in the air may be used to give insights regarding the rise and fall of such concentrations. A graph illustrating each of the contaminants graphed with the x-axis reflecting the number of samples and the y-axis representing the concentration in $\mu g/m^3$.

4.2 Data preparation

The first thing that has to be done is to get the LSTM dataset ready for use with the pollution. In order to do this, the dataset must first be recast as a supervised learning problem, and then the input variables must be normalized. Define the supervised learning task as estimating the pollution level at the current hours *(t)* based on the pollution measurement and the meteorological circumstances from the previous time step.

This formulation is easy to understand and is ideal for demonstrating this point. Consider the following potential alternative formulations:

- Determine the level of pollution that will be present in the air in the next hour based on the meteorological conditions and levels of pollution that have been recorded during the last 24 hours.
- Using the same method as in the previous step, forecast the pollution levels for the next hour based on the "anticipated" weather conditions for the following hour.

After the dataset (a CSV file) has been loaded, the wind direction feature will be label encoded (integer encoded). If you are interested in further investigating this possibility, this may be one-hot encoded at some point in the future. The next step is to normalize each of the features, and after that, the dataset is recast as a supervised learning problem. After that, the weather variables for the hour that will be forecasted *(t)* are taken out of the equation.

4.3 Model fitting

The dataset is divided into training data and test data respectively. After the dataset has been preprocessed, it is input into the model just before to the setting of the network parameters. An optimizer is a crucial component of a neural network that must be configured properly. An optimizer is a technique or group of algorithms that may be used to configure different parameters of neural networks, such as the weights, bias, and learning rates, amongst other things. There are many different optimizers available for neural networks, and which one is used depends on the challenges that are met by the various options.

4.4 Model Evaluation and Error Calculation

As soon as the model has been calibrated, a prediction is generated for the complete test dataset. In this step, we start with the prediction, then combine it with the test dataset, and last, we reverse the scale. In addition to this, we do an inversion of scaling on the test dataset that contains the predicted pollution figures. After converting predictions and actual values back to their respective original scales, we are able to compute an error score for the model. In this circumstance, we compute the Root Mean Squared Error, often known as RMSE, which delivers error in the same units as the variable. The training and testing validation loss graphs are shown in Figure 4.

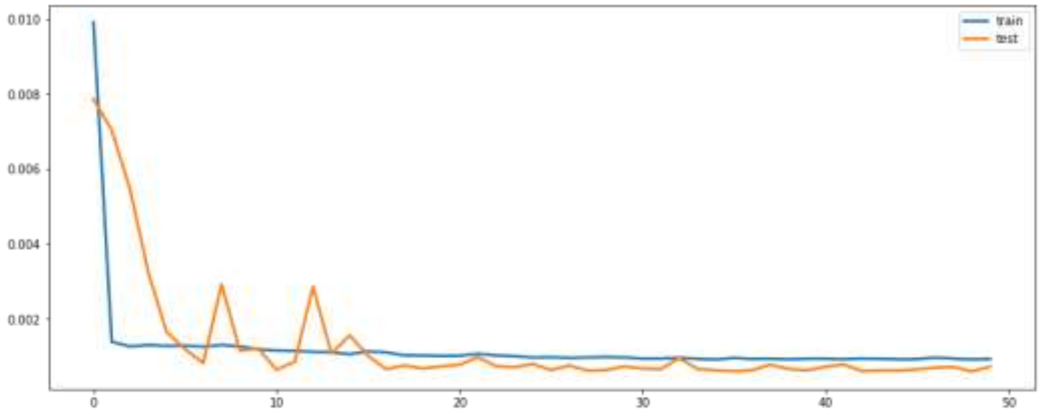
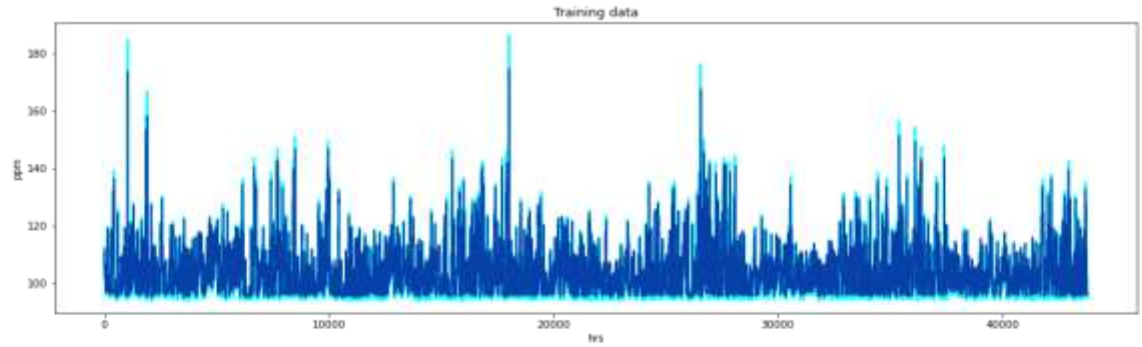
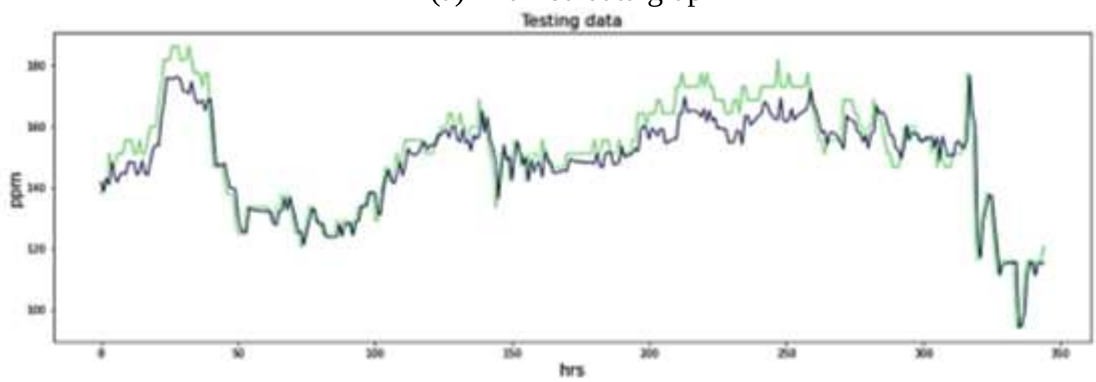


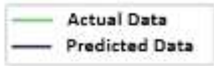
Figure 5: Train and Test Validation graphs of proposed MV-LSTM model

It's interesting to see that our test loss is really lower than our training loss. It's possible that the model is trying to match the training data too well. It's possible that calculating and visualizing RMSE during the course of training may shed further insight on this. After training and testing, the trained and tested data is visualized in Figure 6.



(a) Trained data graph





(b) Testing data graph

Figure 6: Trained and testing data graph

Table 2 shows the actual and predicted values of the proposed model.

Table 2: Actual and predicted values

Actual Value	Predicted Value
164.30016	155.85117
164.30016	159.4245
164.30016	158.52647
168.69308	156.8727
168.69308	159.05635
164.30016	159.13914
164.30016	154.40051
164.30016	155.0527
164.30016	155.488
164.30016	155.85326
173.086	155.98132
173.086	164.1051
177.47891	164.4234
173.086	169.61734
173.086	164.70105

At the conclusion of each training session, both the Train and test loss are estimated. Table 3 shows the comparative analysis. The final RMSE of the model calculated using the test dataset is estimated when the run has been completed.

1. R2 (R-squared) is a metric that measures how well the model fits the data. It is a value between 0 and 1, where 1 indicates a perfect fit and 0 indicates no fit at all.
2. MSE (Mean Squared Error) is a metric that measures the average squared difference between the predicted values and the actual values. It is calculated by taking the sum of the squared differences between the predicted and actual values and dividing by the number of data points.
3. MAE (Mean Absolute Error) is a metric that measures the average absolute difference between the predicted values and the actual values. It is calculated by taking the sum of the absolute differences between the predicted and actual values and dividing by the number of data points.
4. MSLE (Mean Squared Logarithmic Error) is a metric that measures the average squared difference between the logarithm of the predicted values and the logarithm of the actual values. It is useful when the target values have a wide range.
5. RMSE (Root Mean Squared Error) is a metric that measures the square root of the average squared difference between the predicted values and the actual values.

Table 3: Comparative Analysis

Algorithm	R^2	MSE	MAE	MSLE	RMSE
Linear Regression	0.57	615.04	20.64	0.1588	24.8
Logistic Regression	0.61	430.14	16.78	0.0874	20.74
Support vector machine [15]	0.63	366.33	15.24	0.0715	19.14
Random Forest [15]	0.68	239.32	12.95	0.0278	15.47
Convolution Neural Network [16]	0.71	166.66	9.57	0.0084	12.91
Proposed model	0.73	85.44	7.48	0.0035	9.24

Table 3 shows the comparative analysis of the proposed model. Linear Regression produced an R^2, MSE, MAE, MSLE and RMSE of 0.57, 615.04, 20.64, 0.1588 and 24.8 respectively. Logistic Regression produced an R^2, MSE, MAE, MSLE and RMSE of 0.61, 430.14, 16.78, 0.0874 and 0.74 respectively. Support vector machine produced an R^2, MSE, MAE, MSLE and RMSE of 0.63, 366.33, 15.24, 0.0715 and 19.14 respectively. Random Forest produced an R^2, MSE, MAE, MSLE and RMSE of 0.68, 239.32, 12.95, 0.0278 and 15.47 respectively. Convolution Neural Network produced an R^2, MSE, MAE, MSLE and RMSE of 0.71, 166.66, 9.57, 0.0084 and 12.91 respectively. Proposed model produced an R^2, MSE, MAE, MSLE and RMSE of 0.73, 85.44, 7.48, 0.0035 and 9.24 respectively.

Conclusion

Air pollution can be predicted using deep learning techniques, which can automatically learn complex patterns and relationships in the data to make accurate predictions. Deep learning models can be trained on historical air pollution data and other relevant features, such as weather data, traffic patterns, and industrial activities. Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) that is particularly effective for predicting sequential data, such as time series data. LSTM networks can learn complex temporal patterns in the data and can be used to predict future values based on past observations. Multivariate LSTM is a type of LSTM network that can handle input data with multiple variables, where each variable may be observed over time. Unlike univariate LSTM, which takes only one variable as input, multivariate LSTM can model the dependencies between multiple variables, allowing for more accurate predictions. In a multivariate LSTM, each input sequence is a matrix with multiple rows, each representing a different variable, and columns representing time steps. The network processes each time step independently, taking in the current input values for all variables and producing a prediction for each variable at the next time step.

References

[1] Chen, Hui, Brian G. Oliver, Anushriya Pant, Annabel Olivera, Philip Poronnik, Carol A. Pollock, and Sonia Saad. "Effects of air pollution on human health–Mechanistic evidence suggested by in vitro and in vivo modelling." Environmental Research 212 (2022): 113378.
[2] Surakasi, Raviteja, Mohd Yunus Khan, Arif Senol Sener, Tushar Choudhary, Sumantha Bhattacharya, Piyush Singhal, Bharat Singh, and Velivela Lakshmikanth Chowdary. "Analysis of environmental emission neat diesel-biodiesel–algae oil-nanometal additives in compression ignition engines." Journal of Nanomaterials 2022 (2022).
[3] Yang, Liwei, Xiaoqing Gao, Zhenchao Li, and Dongyu Jia. "Quantitative effects of air pollution on regional daily global and diffuse solar radiation under clear sky conditions." Energy Reports 8 (2022): 1935-1948.
[4] Zhao, Xiangyu, Kuang Cheng, Wang Zhou, Yi Cao, and Shuang-Hua Yang. "Multivariate Statistical Analysis for the Detection of Air Pollution Episodes in Chemical Industry Parks." International Journal of Environmental Research and Public Health 19, no. 12 (2022): 7201.
[5] Connolly, Rachel E., Qiao Yu, Zemin Wang, Yu-Han Chen, Jonathan Z. Liu, Ashley Collier-Oxandale, Vasileios Papapostolou, Andrea Polidori, and Yifang Zhu. "Long-term evaluation of a low-cost air sensor network for monitoring indoor and outdoor air quality at the community scale." Science of The Total Environment 807 (2022): 150797.
[6] Liu, Xian, Dawei Lu, Aiqian Zhang, Qian Liu, and Guibin Jiang. "Data-driven machine learning in environmental pollution: Gains and problems." Environmental science & technology 56, no. 4 (2022): 2124-2133.
[7] Sethi, Jasleen Kaur, and Mamta Mittal. "Monitoring the impact of air quality on the COVID-19 fatalities in Delhi, India: using machine learning techniques." Disaster Medicine and Public Health Preparedness 16, no. 2 (2022): 604-611.

[8] Zaini, Nur'atiah, Lee Woen Ean, Ali Najah Ahmed, and Marlinda Abdul Malek. "A systematic literature review of deep learning neural network for time series air quality forecasting." Environmental Science and Pollution Research (2022): 1-33.

[9] Chang, Yue-Shan, Hsin-Ta Chiao, Satheesh Abimannan, Yo-Ping Huang, Yi-Ting Tsai, and Kuan-Ming Lin. "An LSTM-based aggregated model for air pollution forecasting." Atmospheric Pollution Research 11, no. 8 (2020): 1451-1463.

[10] Bui, Tien-Cuong, Van-Duc Le, and Sang-Kyun Cha. "A deep learning approach for forecasting air pollution in South Korea using LSTM." arXiv preprint arXiv:1804.07891 (2018).

[11] Wang, Jingyang, Jiazheng Li, Xiaoxiao Wang, Jue Wang, and Min Huang. "Air quality prediction using CT-LSTM." Neural Computing and Applications 33 (2021): 4779-4792.

[12] Liu, Duen-Ren, Shin-Jye Lee, Yang Huang, and Chien-Ju Chiu. "Air pollution forecasting based on attention-based LSTM neural network and ensemble learning." Expert Systems 37, no. 3 (2020): e12511.

[13] Belavadi, Sagar V., Sreenidhi Rajagopal, R. Ranjani, and Rajasekar Mohan. "Air quality forecasting using LSTM RNN and wireless sensor networks." Procedia Computer Science 170 (2020): 241-248.

[14] Xayasouk, Thanongsak, HwaMin Lee, and Giyeol Lee. "Air pollution prediction using long short-term memory (LSTM) and deep autoencoder (DAE) models." Sustainability 12, no. 6 (2020): 2570.

[15] Liu, Huixiang, Qing Li, Dongbing Yu, and Yu Gu. "Air quality index and air pollutant concentration prediction based on machine learning algorithms." Applied Sciences 9, no. 19 (2019): 4069.

[16] Mao, Yushun, and Shiejue Lee. "Deep convolutional neural network for air quality prediction." In *Journal of Physics: Conference Series*, vol. 1302, no. 3, p. 032046. IOP Publishing, 2019.