International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Air Pollution Prediction using Multivariate LSTM Deep Learning Model

G. Naresh*1, Dr. B. Indira[2]

Submitted: 09/10/2023 **Revised**: 28/11/2023 **Accepted**: 09/12/2023

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, Deep Learning, Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), Time Series.*

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 plays a substantial role in causing various adverse health effects, which encompass respiratory issues, premature mortality, and the necessity for hospitalization due to cardiovascular and respiratory ailments. While air pollution's impact on individuals is considerable, its adverse effects on plants are even more pronounced. This heightened vulnerability primarily stems from prolonged exposure to pollutants, which can lead to harm to the leaves of plants [1]. The chief origins of air pollutants, specifically dust and particulate matter with a diameter less than 10 micrometers (PM10) as well as PM2.5, make up a significant portion of these contaminants. PM2.5 particles, in particular, pose a severe threat due to their smaller size, measuring less than 2.5 microns. These pollutants emanate from stationary sources and emerge as byproducts of both unburned fuel and industrial processes. Furthermore, these same sources are responsible for emitting sulphur dioxide (SO2). 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. Pollutants are categorized based on their level of hazard, following the Indian air quality standard. These Air Quality Indexes (AQI) serve as indicators of the concentrations of key pollutants present in the air. The air quality of a specific region or nation can be regarded as a reflection of the influence exerted by emissions of pollutants in that particular area [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. Nonetheless, comprehending the formation process and mechanism of PM2.5 remains a formidable challenge due to the intricate nature of its characteristics. These attributes, such as their non-linear properties in both time and space, significantly affect the precision of forecasts, rendering the understanding of the mechanism and process quite intricate. Moreover, these characteristics also influence the ability of predictions to accommodate uncertainty effectively. 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

[1]*Research Scholar, Department of Informatics, University College of Engineering, Osmania University, Hyderabad.*
[2]*Associate Professor, Department of MCA, Chaitanya Bharathi Institute of Technology (A), Osman Sagar Rd, Kokapet, Gandipet, Hyderabad.*
Email: bindira_mca@cbit.ac.in
** Corresponding Author Email: nani.naresh2126@gmail.com*

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.

The concept for an Aggregated LSTM model (ALSTM) was proposed by Chang et al. [9]. This model is rooted in the deep learning approach of LSTM. The creators of this distinctive model amalgamate monitoring stations responsible for tracking external sources of pollution, industrial zones in the vicinity, and local area stations to assess air quality. To enhance the accuracy of their early forecasts, the authors employ three distinct LSTM models. These models rely on data from nearby industrial air quality sensors and information from external pollution sources. The authors conducted an evaluation of our pioneering ALSTM model. Multiple evaluation metrics, including MAE, RMSE, and MAPE, were utilized to assess these models, and our innovative ALSTM model outperformed all others in the evaluation.

Recurrent neural networks (RNNs) with long short-term memory units are used by Bui et al. [10]. 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.

A novel approach, known as the CT-LSTM method, has been introduced by Wang et al. [11]. This method integrates the LSTM network model with the chi-square test (CT) to construct the prediction model. The prediction of the Air Quality Index (AQI) level in Shijiazhuang, situated in Hebei province, China. Simple RNN, and the innovative approach detailed in this paper). The outcomes of these five different predictions are subsequently compared.

A novel wind-sensitive attention method has been introduced by Liu et al. [12], utilizing an LSTM neural network model to predict air pollution, specifically PM2.5 concentrations. Subsequently, the variations in spatial-temporal PM2.5 concentrations in nearby sites due to wind direction and speed are considered. Following this initial step, an LSTM neural network is employed for generating initial PM2.5 forecasts based on the pollution levels in the surrounding vicinity. These forecasts are then subjected to "attention." Lastly, to produce secondary PM2.5 predictions, an ensemble learning approach based on eXtreme Gradient Boosting (XGBoost) is utilized. This method combines the initial forecasts with weather predictions.

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, Belavadi et al. [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.

Models for predicting fine PM concentrations were developed by Xayasouk et al. [14] using LSTM and deep autoencoder (DAE) approaches. The model outputs were assessed in terms of root mean square error (RMSE) to evaluate their accuracy. The Internet of Things (IoT), an emerging technology, makes it easy and advantageous to share data with additional devices across wireless networks. However, due of their continual development and technological advancements, IoT systems are more vulnerable to cyberattacks, which could result in strong assaults [17-21]. The fine PM concentrations were accurately projected utilizing the proposed models, with the LSTM model exhibiting slightly superior performance compared to the others.

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.

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 Fig. 1.

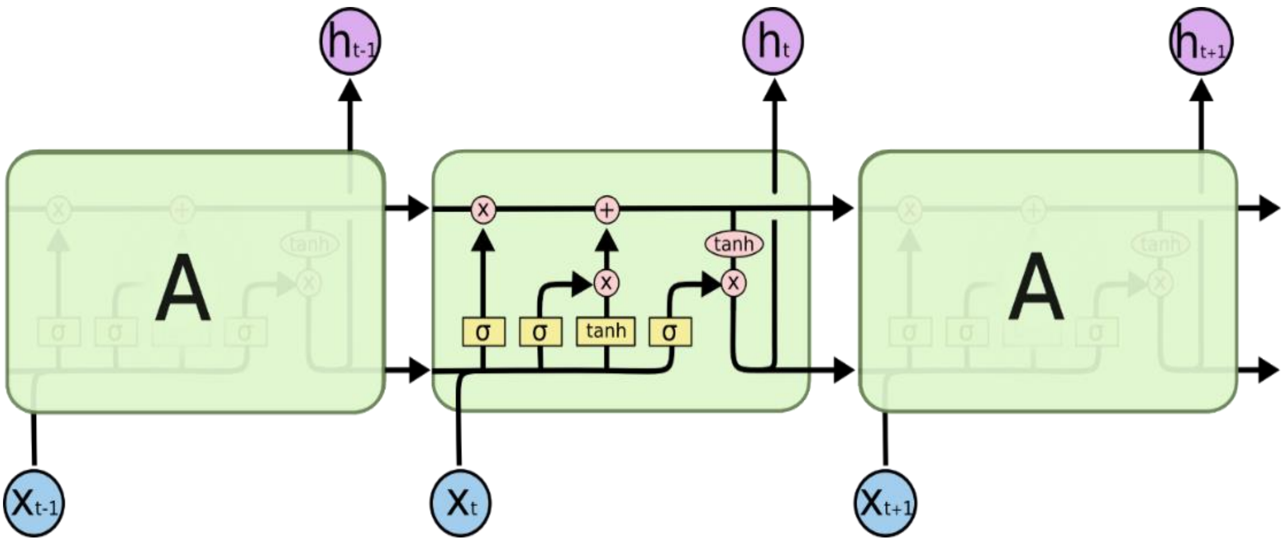


Fig. 1. Basic LSTM Architecture

Input gate: In an LSTM model, the flow of information into the memory cell is regulated by the input gate. This gate is governed by a sigmoid function, which takes into consideration both the present input and the previous hidden state. The sigmoid function produces an output value ranging from 0 to 1, signifying the extent to which information will be permitted to enter the memory cell. When the input gate approaches 0, it indicates minimal or no allowance for information into the memory cell, preserving the prior cell state. Conversely, when the input gate approaches 1, it signifies a substantial information influx into the memory cell, effectively overwriting the previous cell state.

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.

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. The third sigmoid function incorporates values from both the previous hidden state and the currently observed state initially. Following this, the newly generated cell state, derived from the previous cell state, undergoes transformation using the tanh function. Subsequently, element-wise multiplication is performed on these outputs using the multiplier. The network bases its decision regarding the information deemed suitable for the hidden state on the resulting final value. The ability to make accurate predictions hinges on the presence of this implicit condition. Ultimately, both the newly derived cell state and the freshly obtained hidden state are transmitted to the subsequent time step, thereby concluding this section.

Proposed LSTM Architecture

In the process of backpropagation, the vanishing gradient issue is the one that LSTM is mainly designed to address. In the LSTM model, a gating mechanism is employed to control the memorization process, where information can be read, written, and stored through the operation of gates that can open and close as needed. 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.

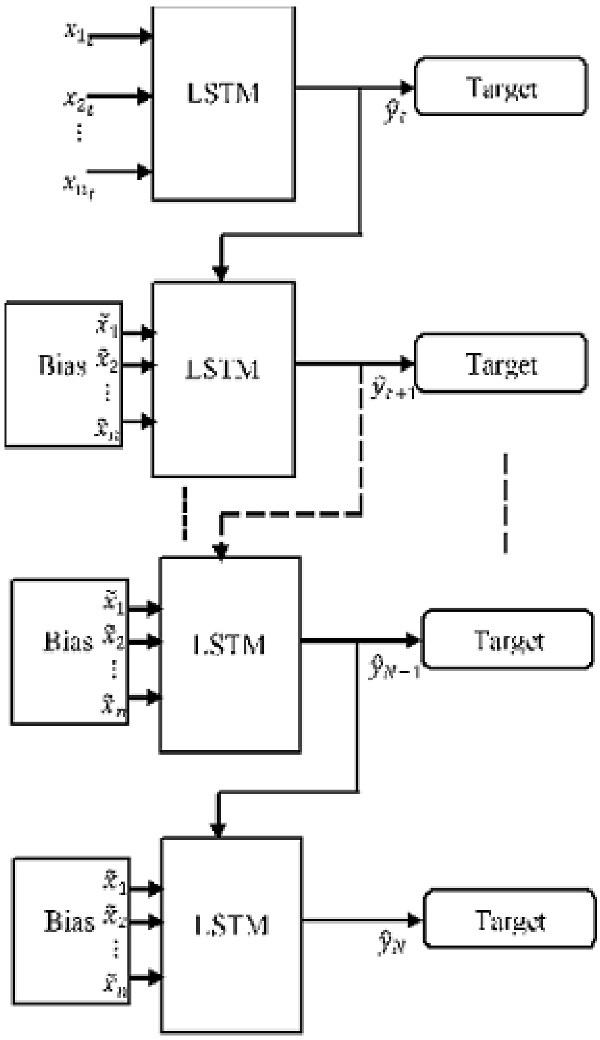


Fig. 2. Proposed MV-LSTM Model

The description of the proposed multivariate LSTM recurrent neural networks may be seen in Fig. 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.

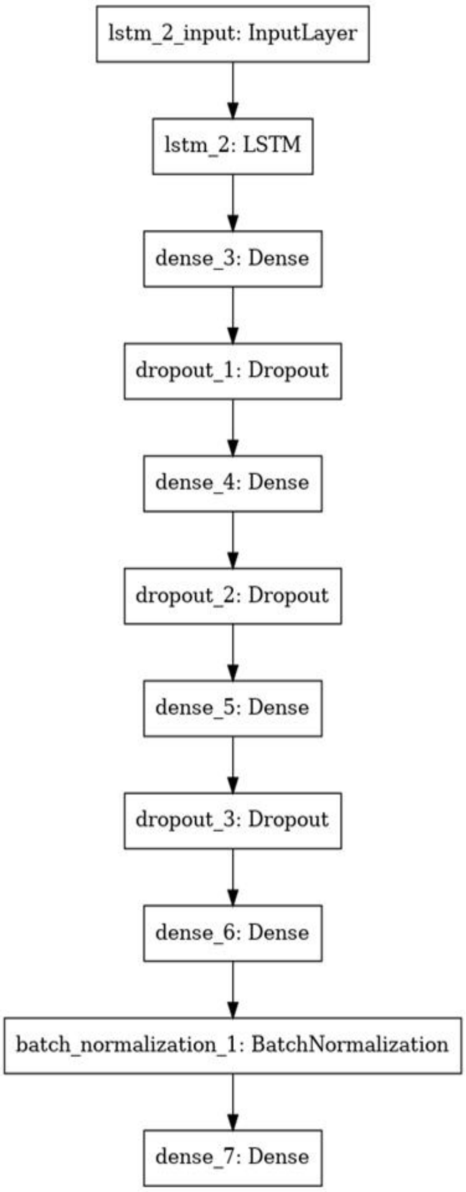


Fig. 3. Proposed Multi Variate LSTM Model

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.

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

Fig. 3 shows the architecture of the proposed multi variate LSTM model. The functionality of the layers is presented here:

- The input layer is where data is initially fed into the neural network. It acts as the interface between the external data and the network's internal layers, transmitting information to subsequent layers.
- The LSTM layer is responsible for handling sequential data, capturing long-range dependencies, and retaining memory of past inputs. It is commonly used in recurrent neural networks (RNNs) for tasks involving sequences, such as natural language processing or time series analysis.
- The Dense layer, also known as a fully connected layer, connects each neuron to every neuron in the previous and subsequent layers. It is responsible for learning complex patterns in the data through weighted connections and activation functions.
- Dropout is a regularization technique used during training to prevent overfitting. It randomly deactivates a fraction of neurons in a layer, reducing co-dependency between neurons and improving the network's generalization ability.
- Batch normalization is a technique that normalizes the activations of a layer by adjusting the mean and variance. It helps stabilize training by reducing internal covariate shift and can accelerate convergence.
- The output dense layer is the final layer of the neural network responsible for producing the network's predictions or outputs. Its architecture depends on the specific task, such as regression or classification, and typically uses activation functions suitable for the task (e.g., sigmoid for binary classification or linear for regression).

4. Simulation Results

The training model and data processing for the proposed MV-LSTM model are described in this section. When training data is transmitted across a network, the primary objective of the training procedure is to minimize any incurred loss, whether it pertains to errors or financial costs resulting from the network's operation. Following the computation of the gradient, which represents the loss concerning a specific set of weights, the weights are subsequently adjusted appropriately. This process is iterated until the optimal weights are determined, leading to the reduction of loss to a minimum level.

There are instances when the gradient approaches near insignificance. It is essential to bear in mind that the gradient of one layer relies on specific attributes of preceding layers. If any of these attributes are substantially small (below 1), the resulting gradient becomes considerably diminished. This diminishment is attributed to what is known as the scaling effect. When this gradient is multiplied by the learning rate, which itself possesses a relatively small value typically within the range of 0.1 to 0.001, it yields a lower value. Consequently, the weight adjustment becomes hardly discernible, resulting in a production outcome that closely resembles the previous state.

When gradients possess large values due to elevated component values, the weights are readjusted to a value surpassing the optimal level. This occurrence is labeled as the "exploding gradients issue." To circumvent this scaling influence, the neural network unit underwent a redesign to maintain the scaling factor at one throughout the entire process.

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 Fig. 4.

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

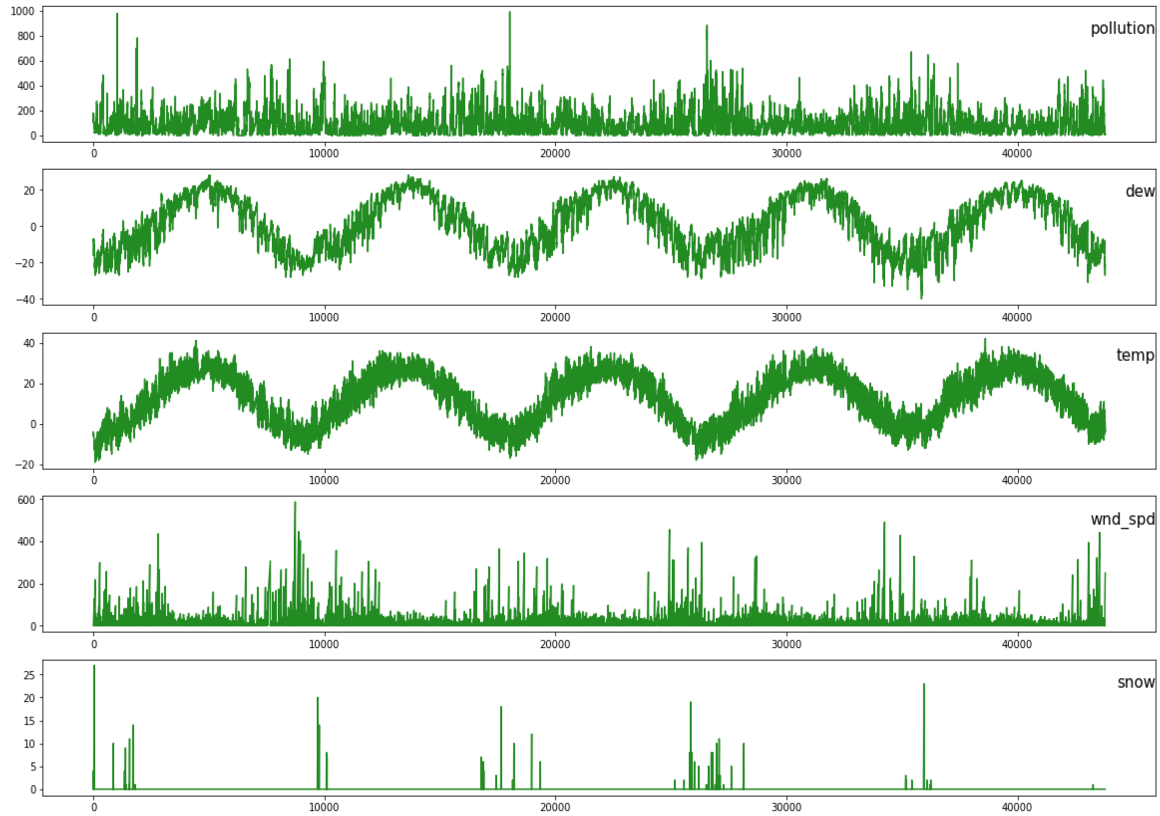


Fig. 4. Data Samples Visualization Graphs

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 $\mu g/m^3$.

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 expression is easily understood and effectively demonstrates the argument presented here. Consider the following alternative phrasings that may be employed:

- Based on the weather conditions and pollution levels recorded over the past twenty-four hours, calculate the anticipated air pollution level for the upcoming hour.
- Utilizing the same procedure as in the previous phase, predict pollution levels for the next hour based on the "expected" weather conditions for that hour.

Following the successful loading of the dataset, which is in CSV format, the wind direction feature is subsequently subjected to label encoding (integer encoding). It is possible that it may undergo one-hot encoding in the future. Should you wish to explore this possibility further, please feel free to inquire. Following this, the dataset is transformed into a supervised learning problem, and the subsequent step involves standardizing each of the features. Subsequently, the weather variables for the predicted hour are excluded from consideration. This particular value is referred to as "t."

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.

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. Furthermore, an inverse scaling operation is performed on the test dataset, encompassing the forecasted pollution data. By reverting the predictions and actual values to their original scales, an error metric for the model can be calculated. In this context, the Root Mean Squared Error (RMSE), recognized for expressing error in the native units of the variable, is computed. The training and testing validation loss graphs are shown in Fig. 5.

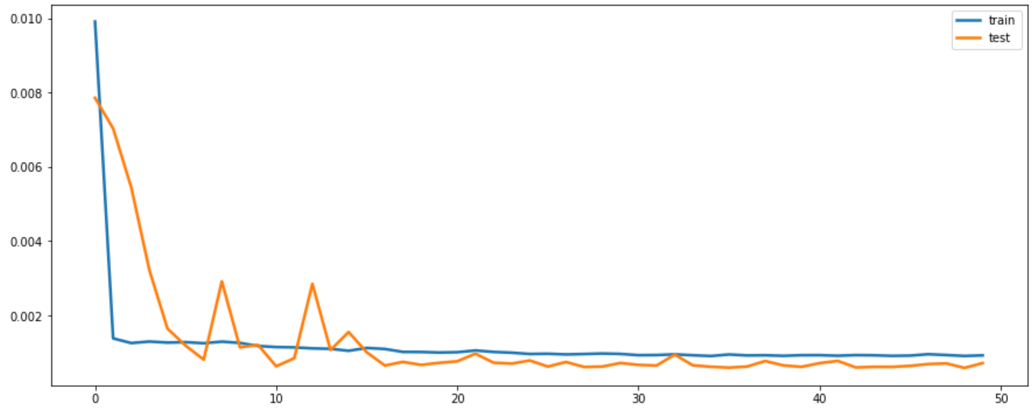
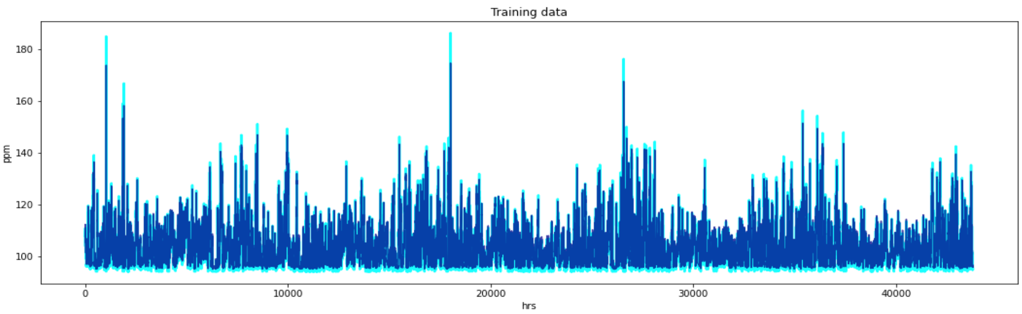
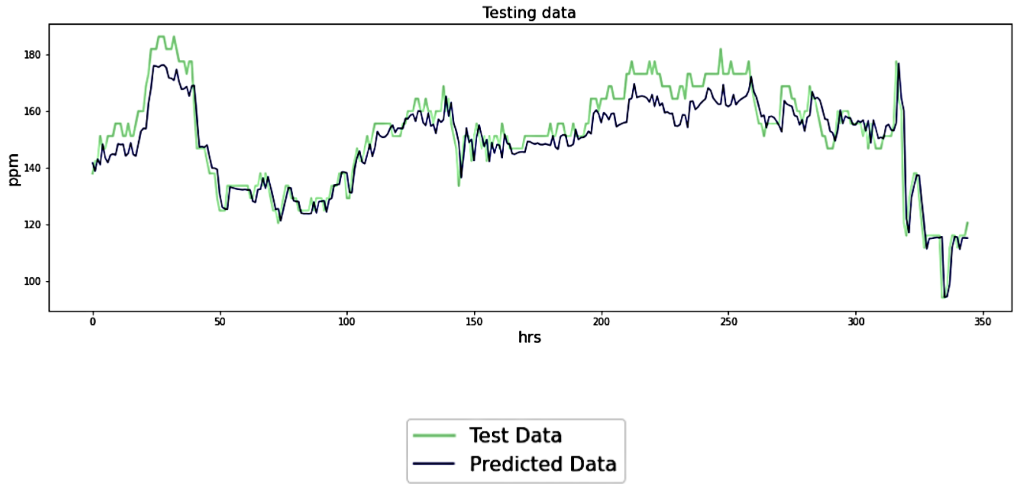


Fig. 5. Train and Test Validation Graphs of Proposed MV-LSTM Model

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 Fig. 6.



(a) Trained Data Graph



(b) Testing Data Graph

Fig. 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. The comparison is performed in terms of R2 (R-squared), MSE (Mean Squared Error), MAE (Mean Absolute Error), MSLE (Mean Squared Logarithmic Error) and RMSE (Root Mean Squared Error).

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.

5. 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] H. Chen, B. G. Oliver, A. Pant, A. Olivera, P. Poronnik, C. A. Pollock and S. Saad, “Effects of air pollution on human health–Mechanistic evidence suggested by in vitro and in vivo modelling,” *Environmental Research*, vol. 212, pp. 1-64, 2022.

[2] R. Surakasi, M. Y. Khan, A. S. Sener, T. Choudhary, S. Bhattacharya, P. Singhal, B. Singh and V. L. Chowdary, “Analysis of environmental emission neat diesel-biodiesel–algae oil-nanometal additives in compression ignition engines,” *Journal of Nanomaterials*, pp. 1-7, 2022.

[3] L. Yang, X. Gao, Z. Li and D. Jia, “Quantitative effects of air pollution on regional daily global and diffuse solar radiation under clear sky conditions,” *Energy Reports*, vol. 8, pp. 1935-1948, 2022.

[4] X. Zhao, K. Cheng, W. Zhou, Y. Cao and S. H. Yang, “Multivariate Statistical Analysis for the Detection of Air Pollution Episodes in Chemical Industry Parks,” *International Journal of Environmental Research and Public Health*, vol. 19, no. 12, pp. 1-21, 2022.

[5] R. E. Connolly, Q. Yu, Z. Wang, Y. H. Chen, J. Z. Liu, A. Collier-Oxandale, V. Papapostolou, A. Polidori and Y. 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*, vol. 807, pp. 1-11, 2022.

[6] X. Liu, D. Lu, A. Zhang, Q. Liu and G. Jiang, “Data-driven machine learning in environmental pollution: gains and problems,” *Environmental science & technology*, vol. 56, no. 4, pp. 2124-2133, 2022.

[7] J. K. Sethi and M. 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*, vol. 16, no. 2, pp. 604-611, 2022.

[8] N. A. Zaini, L. W. Ean, A. N. Ahmed and M. A. Malek, “A systematic literature review of deep learning neural network for time series air quality forecasting,” *Environmental Science and Pollution Research*, pp. 1-33, 2022.

[9] Y. S. Chang, H. T. Chiao, S. Abimannan, Y. P. Huang, Y. T. Tsai and K. M. Lin, “An LSTM-based aggregated model for air pollution forecasting,” *Atmospheric Pollution Research*, vol. 11, no. 8, pp. 1451-1463, 2020.

[10] T. C. Bui, V. D. Le and S. K. Cha, “A deep learning approach for forecasting air pollution in South Korea using LSTM,” *arXiv preprint arXiv:1804.07891*, pp. 1-6, 2018.

[11] J. Wang, J. Li, X. Wang, J. Wang and M. Huang, “Air quality prediction using CT-LSTM,” *Neural Computing and Applications*, vol. 33, pp. 4779-4792, 2021.

[12] D. R. Liu, S. J. Lee, Y. Huang and C. J. Chiu, “Air pollution forecasting based on attention-based LSTM

neural network and ensemble learning," *Expert Systems*, vol. 37, no. 3, pp. 1-16, 2020.

[13] S. V. Belavadi, S. Rajagopal, R. Ranjani and R. Mohan, "Air quality forecasting using LSTM RNN and wireless sensor networks," *Procedia Computer Science*, 2020, vol. 170, pp. 241-248.

[14] T. Xayasouk, H. Lee and G. Lee, "Air pollution prediction using long short-term memory (LSTM) and deep autoencoder (DAE) models," *Sustainability*, vol. 12, no. 6, pp. 1-17, 2020.

[15] H. Liu, Q. Li, D. Yu and Y. Gu, "Air quality index and air pollutant concentration prediction based on machine learning algorithms," *Applied Sciences*, vol. 9, no. 19, pp. 1-9, 2019.

[16] Y. Mao and S. Lee, "Deep convolutional neural network for air quality prediction," *Journal of Physics: Conference Series*, vol. 1302, no. 3, pp. 1-6, 2019.

[17] M. Yaseen, H. S. Salih, M. Aljanabi, A. H. Ali and S. A. Abed, "Improving Process Efficiency in Iraqi universities: a proposed management information system," *Iraqi Journal For Computer Science and Mathematics*, vol. 4, no. 1, pp. 211-219, 2023.

[18] M. Aljanabi and S. Y. Mohammed, "Metaverse: open possibilities," *Iraqi Journal For Computer Science and Mathematics*, vol. 4, no. 3, pp. 79-86, 2023.

[19] A. S. Shaker, O. F. Youssif, M. Aljanabi, Z. Abbood and M.S. Mahdi, "SEEK Mobility Adaptive Protocol Destination Seeker Media Access Control Protocol for Mobile WSNs," *Iraqi Journal For Computer Science and Mathematics*, vol. 4, no. 1, pp. 130-145, 2023.

[20] H. S. Salih, M. Ghazi and M. Aljanabi, "Implementing an Automated Inventory Management System for Small and Medium-sized Enterprises," *Iraqi Journal For Computer Science and Mathematics*, vol. 4, no. 2, pp. 238-244, 2023.

[21] G. Perumal, G. Subburayalu, Q. Abbas, S. M. Naqi and I. Qureshi, "VBQ-Net: A Novel Vectorization-Based Boost Quantized Network Model for Maximizing the Security Level of IoT System to Prevent Intrusions," *Systems*, vol. 11, no. 8, pp. 1-25, 2023.