DEEP LEARNING-BASED AIR POLLUTION PREDICTION IN SMART CITIES

A PROJECT REPORT

Submitted by

SHALIMA SHAJAHAN
(TKM20MCA2035)

to

The APJ Abdul Kalam Technological University

In partial fulfillment of the requirements for the award of the degree of

MASTER OF COMPUTER APPLICATIONS

Thangal Kunju Musaliar College of Engineering
Kerala

DEPARTMENT OF COMPUTER APPLICATIONS

JULY 2022

DECLARATION

I undersigned hereby declare that the project report on '**DEEP LEARNING BASED AIR POLLUTION PREDICTION IN SMART CITIES**" , submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Prof.Dr.Fousia M Shamsudeen. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Kollam

18/07/2022 Shalima Shajahan

DEPARTMENT OF COMPUTER APPLICATIONS

TKM COLLEGE OF ENGINEERING

CERTIFICATE

This is to certify that, the project report entitled "**DEEP LEARNING BASED AIR POLLUTION PREDICTION IN SMART CITIES**" submitted by **SHALIMA SHAJAHAN (TKM20MCA2035)** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the degree of Master of Computer Applications, is a bonafide record of the project work carried out by her under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Internal Supervisor	Head of the Department	External Examiner

ACKNOWLEDGEMENT

First and foremost, I thank GOD almighty and my parents for the success of this project. I owe sincere gratitude and heart full thanks to everyone who shared their precious time and knowledge for the successful completion of my project.

I express my sincere gratitude to **Dr. T A Shahul Hameed**, Principal of T.K.M College of Engineering for giving me an opportunity to present my project. With a profound sense of gratitude, I would like to express my heartfelt thanks to **Dr. Fousia.M.Shamsudeen**, Head of the Department, for providing me with best facilities and motivated me throughout the work of my project.

I profusely thank all other faculty members in the department and all other members of TKM College of Engineering, for their guidance and inspirations throughout my course of study. I owe my thanks to my friends and all others who have directly or indirectly helped me in the successful completion of this project.

SHALIMA SHAJAHAN

ABSTRACT

One of the most significant issues affecting urban living is air pollution, a form of environmental hazard. One of the air pollutants responsible for a number of illnesses is PM2.5, or Particulates with a diameter of less than 2.5 mm. Therefore, it is crucial to accurately forecast PM2.5 concentrations to protect the public from possible negative consequences of air pollution in advance. It is now feasible to anticipate quality of air in cities using deep learning approach because to the quick advancement of deep learning technology and their use in all facets in life. In the context of smart cities, this study examines experiments including deep learning techniques. Using a deep learning solution, CNN Bi-LSTM, with a spacio temporal feature, and historical features on pollutants, weather, and PM2.5 concentration at nearby stations, it is possible to predict the hourly prediction of air quality in India, especially Delhi. Additionally comparing the performance differences across several algorithms like LSTM, Bi-LSTM, GRU, Bi-GRU, CNN LSTM, and CNN-Bi-LSTM model. The experiment's findings indicate that the mean absolute error of CNN Bi-LSTM and the RMSE both experience drops of 0.20224 and 0.33256, respectively. CNN Bi-LSTM gives the greatest results for air pollution prediction when compared to Bi-LSTM and LSTM, and its training time is cut by 12.5 and 50.9 cent, respectively. . The primary conclusions of this study are as follows, based on performance assessment and result comparison: this model can successfully capture the spatial and time aspects of the data using CNN and Bi-LSTM, with more precisely and has stability.

.

CONTENTS

LIST OF FIGURES

List of Tables

CHAPTER 1

INTRODUCTION

Increasing air pollution has become a key worry for urban environmental quality of life. With the economy's rapid expansion, industrial activities are expanding more often, resulting in a quicker rate of pollution. This pace is growing and, if not controlled, can have negative consequences for humans and other living species. Environment pollution is a serious problem that affects all humans and all living forms, with pollution created by industry playing a significant role. Industrial pollution is caused by companies and other industrial operations that spew hazardous byproducts and trash into the environment. Recent studies depict the depressing reality of air pollution and its health hazards in cities throughout the world. According to data from the Urban Population website, 55.714 percent of individuals moved to cities in 2019. This population growth would result in a number of health, transportation, and air quality issues. WHO estimates that seven million people die each year as a result of air pollution since research indicated that 90% of people worldwide breathe air that is more polluted than what is considered safe by WHO criteria.

The rise in industrialization and the resulting increased haze has drastically increased environmental pollution during the past few decades. What's more, it gets worse than ever before. PM2.5, or particulate matter with a size of 2.5 m or smaller, is one of the dangerous pollutants. Such particles cause substantial health harm. The economies are at risk due to the bad air quality, in addition to people's health and lives. In addition to having an impact on people and animals, air pollution also directly affects plants. Long-term pollution exposure has an impact on plants because the component of the plant most vulnerable to pollution, the leaves, suffer damage. Because of this, gathering data on air quality is crucial for taking preventative action, but gathering and evaluating this data is still a challenging undertaking. To produce data analysis that is very accurate and efficient.

The effectiveness of an air pollution monitoring and forecasting system is crucial for both human health and government decision-making. Therefore, it necessitates careful attention. The Air Quality Index (AQI), a color-coded indicator designed by the United States Environmental Protection Agency to assess and forecast daily air quality, was established. This indicator examines the connection between a collection of contaminants that are believed to have a major adverse effect on individuals., including ozone , PM2.5, PM10,CO, NO2, and SO2. The health impacts of these contaminants, which may manifest hours or days after someone breathes them in, are the main focus of AQI. The pollution is measured using an AQI scale that ranges from 0 to 500. The dangers associated with air pollution rise in direct proportion to the AQI values. Pollutants can harm your health over time by causing conditions including COPD (Chronic Obstructive Pulmonary Disease), coughing, wheezing, asthma, and respiratory illnesses. In terms of the long-term effects of exposure to air pollution, this exposure may result in more serious health problems such chronic asthma, pulmonary insufficiency, and cardiovascular disorders, among others.

Recurrent Neural Networks (RNN) and their variations are among the numerous deep learning algorithms that have been created in response to the growing popularity of artificial intelligence. The most popular model in air quality forecasting, long short-term memory (LSTM), takes into account the time dependencies of a frequent phenomena seen in the PM2.5 concentration sequence. The intricacy of PM2.5 generation makes it necessary to design a model with high accuracy and predictive efficiency in order to effectively anticipate PM2.5 concentration. Therefore, advise evaluating multivariate deep learning models using a variety of criteria. To that purpose, this work aims to conduct an investigation with algorithms like long shot term memory, gated recurrent unit, bidirectional long short term memory, bidirectional GRU, convolutional LSTM and convolutional bidirectional long short term memory. As a result, the research attempts to uncover a differentiation between the outcomes produced by various methodologies in order to understand how well they forecast PM2.5 concentration. Additionally, using weather information and the concentration of nearby stations, our research seeks to create a PM2.5 forecasting model with high accuracy.

A method for PM2.5 prediction was developed in this study using cutting-edge deep neural networks. This led to the suggestion of a Convolutional BiLSTM estimating model. In this work, six other predictive deep learning models were created for comparison with the one we proposed.

1.1 PROBLEM STATEMENT

- Information on pollution concentrations in nearby locations cannot be used by any of the models. Because pollutants can spread to other places, changes in particles are related to both time and geography. Spatial information must be taken into consideration.
- A more precise forecasting model for air quality is required for early warning and air pollution control management. The effects of featured states at various times in the past on future prediction are, however, ignored by the current methods for PM2.5 concentration and air quality prediction. With this in mind, the system will also receive meteorological data as input in addition to air pollution concentrations..

1.2 OBJECTIVE

The main goal of this research is to create a CNN Bi-LSTM model for predicting air pollution, assess how well each major deep learning approach performs at doing so, and identify the model that predicts air pollution the most correctly. The other sections of the paper are as follows: section II presents a review of the literature; section III a methodology; section IV results and discussion; and section V a conclusion.

CHAPTER 2

LITERATURE SURVEY

A complete review of the literature that relates to a given topic is known as a literature review. When using a literature review, research questions are first identified, after which one strives to provide a solution by looking up and assessing pertinent material. The ability to re-analyze study results can lead to the development of fresh insights, which is one benefit of literature reviews. A literature review summarizes and explains the whole and most recent body of information about a subject that may be found in scientific books and journal articles. There are two types of literature assessment that students may be asked to write at university: one is done as a separate assignment for a course, and the other is written as part of an introduction to or preparation for a larger work, typically a thesis or research report. Nevertheless of the type of review you are composing, your topic, point of view, and type of hypothesis or thesis argument will all be determined. Reading published research articles or the introductory sections of theses and dissertations in your own field of study might help you better comprehend the distinctions between these two forms. Examine the manner they address the issues and the organization of their arguments.

2.1 PURPOSE OF THE LITERATURE REVIEW

1. It makes it simple for readers to conduct research on a certain subject by choosing excellent articles or studies that are pertinent, significant, vital, and valid and compiling them into a single comprehensive report.

2. It forces researchers to sum up, assess, and contrast original research in that particular field, giving them a great place to start when conducting research in a new field.
3. It makes sure that previous study is not repeated by researchers.
4. It can suggest topics to focus on or give hints as to where future research is going.
5. It emphasises the important findings.
6. It points out gaps, discrepancies, and inconsistencies in the literature.
7. It offers a helpful critique of the methods and strategies used by other researchers.

2.2 RELATED WORKS

Researchers have developed a number of diverse ways for predicting PM2.5 levels based on statistical models and methods of machine learning. Deep neural networks have just recently been adopted for the purpose of predicting pollutant concentrations. Deep learning may be able to address problems because it employs a greater number of datas, and it processes every levels togetherly to generate best précised findings . Because of these beneficial qualities, deep learning is an approach that is well-suited for forecasting air .Garg S ,Jindal H et.al [1] uses models like ARIMA, FB Prophet , LSTM, and CNN to look at and predict the PM2.5 values at twelve stops in Beijing. With weather, air quality, and weather forecasting data from the past. LSTM performed better than all other models, according to most stations. The MAE is 13.2, and the RMSE is 20.

Xayasouk T, Lee H .et.al, Lee G.et.al [2] propose a model that can predict PM concentration at 25 monitoring stations in Seoul, South Korea using denoising Autoencoders . The model is based on PM2.5 concentration and weather data. In terms of future prediction, the results show that the LSTM model performed better than the DAE model..

Jeya S , Sankari L.et.al [3] create a BI-LSTM model to predict PM2.5 concentration in China. The weather and PM2.5 concentration were taken from hourly data from the US Embassy for Beijing. Here's how well the advanced model worked: The MAE is 7.53, the RMSE is 9.86, and the SNAPE is 0.1664. Researchers [4] One may calculate the PM2.5 contamination of stations in Beijing using past aqi, meteorological parameters, weather prediction records, and data on the day of the month. They achieved this utilising an artificial neural network and long short-term memory that is completely linked, with MAE of 23.97 and RMSE of 35.82 over 1-6hrs, they demonstrated that the long short term memory with fully connected model performs better than LSTM model and the ANN model.

Rijal N, Gutta RT et.al. [5], To calculate PM2.5 concentrations from photos taken outside, the scientists suggested to use a hybrid model. The basic learners for three neural networks include VGG-16, Resnet50 and Inception-v3.The results of the experiments showed that suggested ensemble can provide a more accurate evaluation of PM2.5 than the three separate deep learning networks employed. Processing geographical data using CNN has shown to be effective. Additionally, using an analysis of satellite photos, this technique has been utilised to determine the concentration of contaminants in metropolitan areas [6, 7]. But occasionally, monitoring data such as wind speed, temperature, and location are the only available data instead of images.

Al-Janabi, S., Mohammad, M et.al.[9] Smart Air Quality Prediction Model, was put out by a group of Iraqi researchers and operations are done using particle swarm optimizer ,which uses the LSTM algorithm and unsupervised learning. This model forecasts six pollutant concentrations for the following two days. The information is collected from thirty five sites across China. In order to build the LSTM model and PSO, the data must first be gathered and the missing values must be handled. The forecast over a 48-hour period was well evaluated using the Symmetric Mean Absolute Percentage Error, which was used. The model is new and used new methodologies.

Ibrahim Kok,Suat Ozemir.et.al. [10] employed an LSTM model to forecast PM25 and NO2 levels for future air quality in a smart city in 2017. Support Vector Regressor was used concurrently with LSTM to compare the results.. They start by constructing the best LSTM model by experimenting with different hyperparameters. The dataset that was used contained 17568 samples with eight characteristics (O3,CO,NO2 particulate matter (PM2.5, PM10), SO2, longitude, latitude, and timestamp) that were gathered at five-minute intervals from IOT devices in smart cities. They employed evaluation metrix such as RMSE and MAE to assess performance. Results from SVR were better than those from LSTM. Therefore, the LSTM and SVR had accuracy rates of 95 and 92.95 percent, respectively. Due to its long-term memory, the

LSTM model ultimately outperforms other models in terms of accuracy and predictability even though it has a straightforward network topology.

Bui, T. C et.al. [12] using LSTM and RNN several South Korean researchers attempted to forecast PM2.5 levels in 2018.. In order to boost accuracy, this model enables the stacking of several RNN layers. This model was applied to two sets of data, each containing 30 million records from different sources between 2008 and 2018, which were then split into 27.936 rows for the purposes of training and testing the model. Twenty four encoding and eight decoding timesteps are used with two layers of RNN, the RMSE for RAE and RMSE is 12.41 and 13.47, respectively. The findings of two RNN are 12.85 and 12.79 in RMSE utilising encoding decoding model from past 24 hours.

Chang, Y. S. et al. [13]. In this study, an aggregated LSTM model was utilised to forecast PM2.5 constrictions in Taiwan over the next 1–8 hours. These fortyone aggreegated models shall produce the output with combining the forecasted characteristics. Aggregated ALSTM model has two categories and is equivalent to voting in that way. The two types are aggregation-learning and mixing. The first kind of "mixing" is training a separate model to provide predictive traits, which are then used to instruct a framework to blend them into a forecasted output data. The group of aggregation training is also in charge of synchronising the sub-models' predictive traits and producing a final composite dataset. The results showed that the ALSTM model may boost forecasting precision when compared to the SVR, GBTR, and LSTM utilising various assessment metrics as MAE, RMSE, and MAPE.

Le, S. K. et.al.[14] In order to predict the air quality, they used CNN to treat Seoul, Korea, as a single image and interpolate and anticipate the city's air quality. They then employed the spatiotemporal data (ST), which also offers indicators for the quality of the outside air as well as data on the weather, traffic volume, and average driving speed. These spatiotemporal data were employed in the Convolutional Long Short-Term Memory, the architecture that has proven to be superior to other structures and is the one that was suggested.

Sharma, E. et al. [15] To predict Total Suspended Particulate (TSP) for the upcoming hour. The mass of PM particles in the atmosphere is gauged by TSP. By combining CNN with the LSTM network, they created a hybrid CLSTM model. By utilising six assessment metrics and comparisons with various machine learning techniques (R, RMSE, MAE, WI, L, ENS).

Bekkar et.al [16] used different deep learning architectures to predict the concentration of air pollution especially particles having diameter less than 2.5 micrometre.Author proposed a convolutional long short term model and compare it with other models such as GRU,Bi-GRU,LSTM, bidirectional LSTM ,convolutional neural networks.By using three evaluation metrics and comparison showed proposed model has the best accuracy .

CHAPTER 3

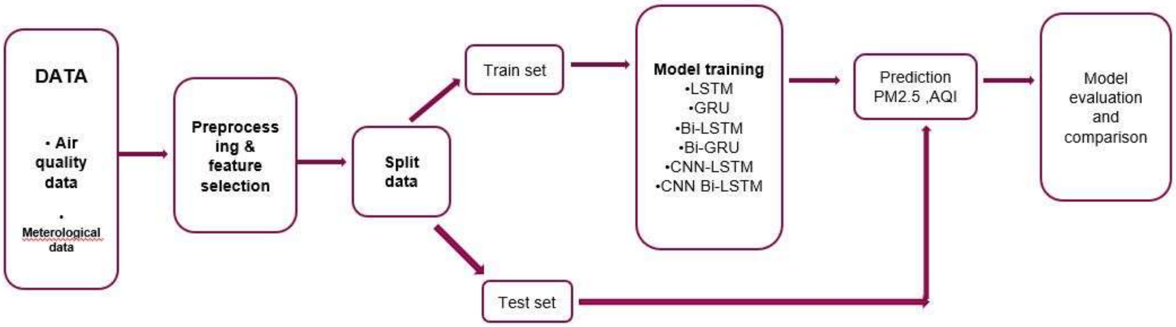
METHODOLOGY

3.1 PROPOSED SYSTEM

The input variables for this study comprise meteorological information, nearby stations, and pollution components throughout various time periods. After preparing the data by normalizing and standardizing the data, selecting features based on the connection between the features and PM2.5 concentration. Correlations between space and time are intricate and extensive. In our study, the model is built using past data from the target station and nearby stations that have been combined with additional characteristics. According to the findings, the suggested combination outperforms the competition in terms of PM2.5 and AQI prediction accuracy as well as effectiveness in extracting spatiotemporal variables. 2. The spatial-temporal properties of the data are retrieved using the suggested model. Integrating the benefits of the CNN model's convolutional neural network, which is efficient at filtering

3.2 SYSTEM ARCHITECTURE

The system architectural design is used to abstract the overall outline of the software system and the relationships, constraints, and boundaries between components. It is an important tool as it provides an overall view of the physical deployment of the software system. This study uses several deep learning models for prediction of air quality data. The figure 3.1 below shows system architecture. The proposed system consist of three major phases: Data preprocessing ,training and testing the data set and evaluation of the model



Workflow for predicting pm2.5 and AQI concentration

3.3 MATERIALS AND METHODS

1. DATA COLLECTION

From several air quality monitoring sites in Delhi, we gathered hourly data from June 2020 to November 2021.Dataset chosen in this article consists of 12237 instances and 17 attributes. In this dataset, the air quality and pollutant concentrations at 47 locations are shown.. The presented method was implemented on two datasets collected from multiple resources. The statistics contain PM10, PM2.5, NO2, CO, O3 & SO2 as air pollutants. The data set also includes meteorological information from the same time period, such as weather conditions ,wind speed, moisture, temperature and pressure. The measurement for air pollution is g/m3. The most common weather kinds are bright, snowy, overcast, light and heavy rain, and blowing sand. Temperature is measured in degrees Celsius. Hectopascals is the name of the unit of atm pressure (hPa). Water vapor proportion in the air is referred to as humidity in percentage. Meters per second (m/s) is the measurement of wind speed. This data set mainly crawled from Meteostat and from OpenAQ The Weather's Record Keeper Meteostat is one of the largest vendors of open weather and climate data. The platform provides access to long-term time series of thousands of weather stations. OpenAQ is a non-profit organization empowering community around the globe to clean their air by harmonizing, sharing, and using open air quality data and also it is a real-time, interactive air quality monitoring website that enables visitors to find out about air quality monitoring and levels in their local area. Both data are collected based on the location ID, date and time.

2. PREPROCESSING

2.1 Filling The Missing Values

The 12237 entries in this collection have many attributes in each station. Date, PM2.5 and PM10 concentrations, Sulfur Dioxide, Nitrogen Dioxide, Carbon Monoxide, Ozone, Relative Humidity, Temp, Air Pressure, Cumulated Airspeed, Cumulated Hour of Snowfall, and Cumulated Hours of Rainfall make up the data. However, the equipment used to monitor the air quality and the weather will result in data leakage owing to equipment failure for various unavoidable causes. Data mining may be affected to some extent by the presence of such missing variables. The most often used methods for replacing missing field values in time-independent (non-chronological) data are the mean and median values. This is not the case, though, when it comes to time series. Many imputation approaches are used to tackle problems with incomplete data. According to a research, the optimum approach for estimating daily monitoring information for PM10 for each and every percent of generated incomplete data is linear interpolation.. The equation for linear interpolation is as follows:

$$y - y_1 = \frac{(y_2 - y_1) \times (x - x_1)}{x_2 - x_1} \tag{1}$$

2.2 **Normalizing The Data Set**

Since their measurement units differ, the value ranges of the various air quality data components also varied significantly. To remove the effects of various numerical ranges, the data should be normalised. Additionally, normalising the data collection can increase prediction accuracy and speed up training. Below is a display of the original sample data's normalising procedure. Where X is the actual data, min denotes the original data's least value, and max denotes the actual data's highest value. The distribution of each element in the processed data set will be between [0, 1]. The training data set and test data set are separated from the normalised data set in an 8:1 ratio.

$$x = \frac{x - min}{max - min} \tag{2}$$

3.3 Feature selection

The process of feature selection, which may be carried out in a variety of methods, is crucial to machine learning applications. To determine the relationship between the dependent and independent variables, the majority of prior research relied on a mathematical relationship. Developing a connection between the goal real output and those attributes decreases the complexity of training and when multiple characteristics are added to the network for training, performance is improved. The Pearson correlation is the method that is most frequently used to establish the link between two variables. Its coefficient may be calculated using the following equation.

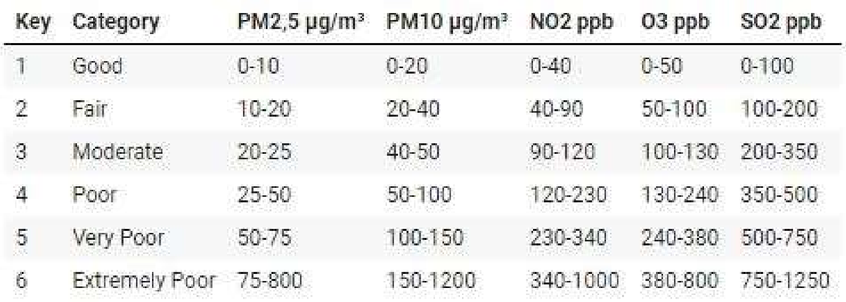
$$\gamma = \frac{\sum_{i=1}^{n}(x_1-\pi)(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-x)^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}} \qquad (3)$$

Air quality data

We can identify a variety of contaminants in the atmosphere, and when their concentrations rise, the air's quality suffers. We examined the connections between the air quality characteristics and discovered a strong relationship between PM2.5, PM10, and CO. The 24-hour exposure's levels of health concern (i.e., hazardous, mild, acceptable, etc.) are determined by the AQI scale. The US-EPA 2016 standard is represented by the Air Quality Index scale in Table S.1. According to Wu et al. (2020), the equation below is used to translate concentrations into AQI:

$$\frac{I=(I_h-I_l)\times(C-C_l)+I_l}{C_h-C_l} \qquad (4)$$

Where I stands for "(Air Quality) Index," C for "Pollutant Concentration," C_{low} for "Concentration Breakpoint That Is C," C_{high} for "Concentration Breakpoint That Is C," and I_{low} and I_{high} for "Index Breakpoint That Is C."



Key	Category	PM2,5 µg/m³	PM10 µg/m³	NO2 ppb	O3 ppb	SO2 ppb
1	Good	0-10	0-20	0-40	0-50	0-100
2	Fair	10-20	20-40	40-90	50-100	100-200
3	Moderate	20-25	40-50	90-120	100-130	200-350
4	Poor	25-50	50-100	120-230	130-240	350-500
5	Very Poor	50-75	100-150	230-340	240-380	500-750
6	Extremely Poor	75-800	150-1200	340-1000	380-800	750-1250

Table 1: Air Quality index

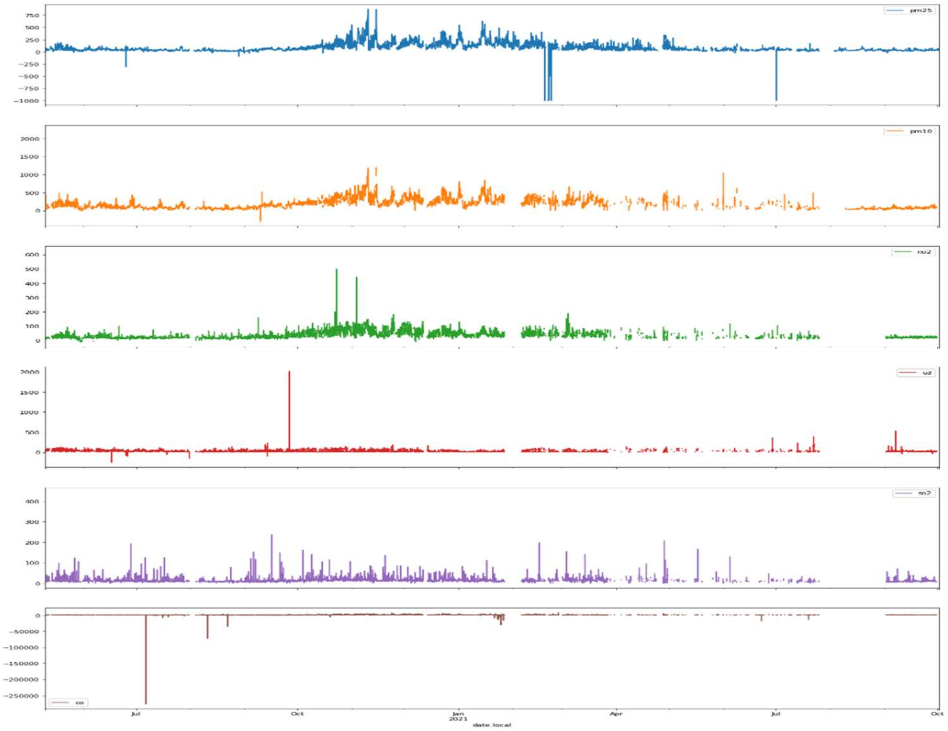


Fig 2: Distribution of different variables of air quality data

Meteorological features

Air quality is impacted by weather variables such as air pressure, temp, and humidity levels. For instance, heavy winds speed will lower PM2.5 concentrations, high humidity will usually make air pollution worse, and high air pressure will usually make the air quality better. Therefore, for the purpose of forecasting air quality, meteorological factors are crucial.

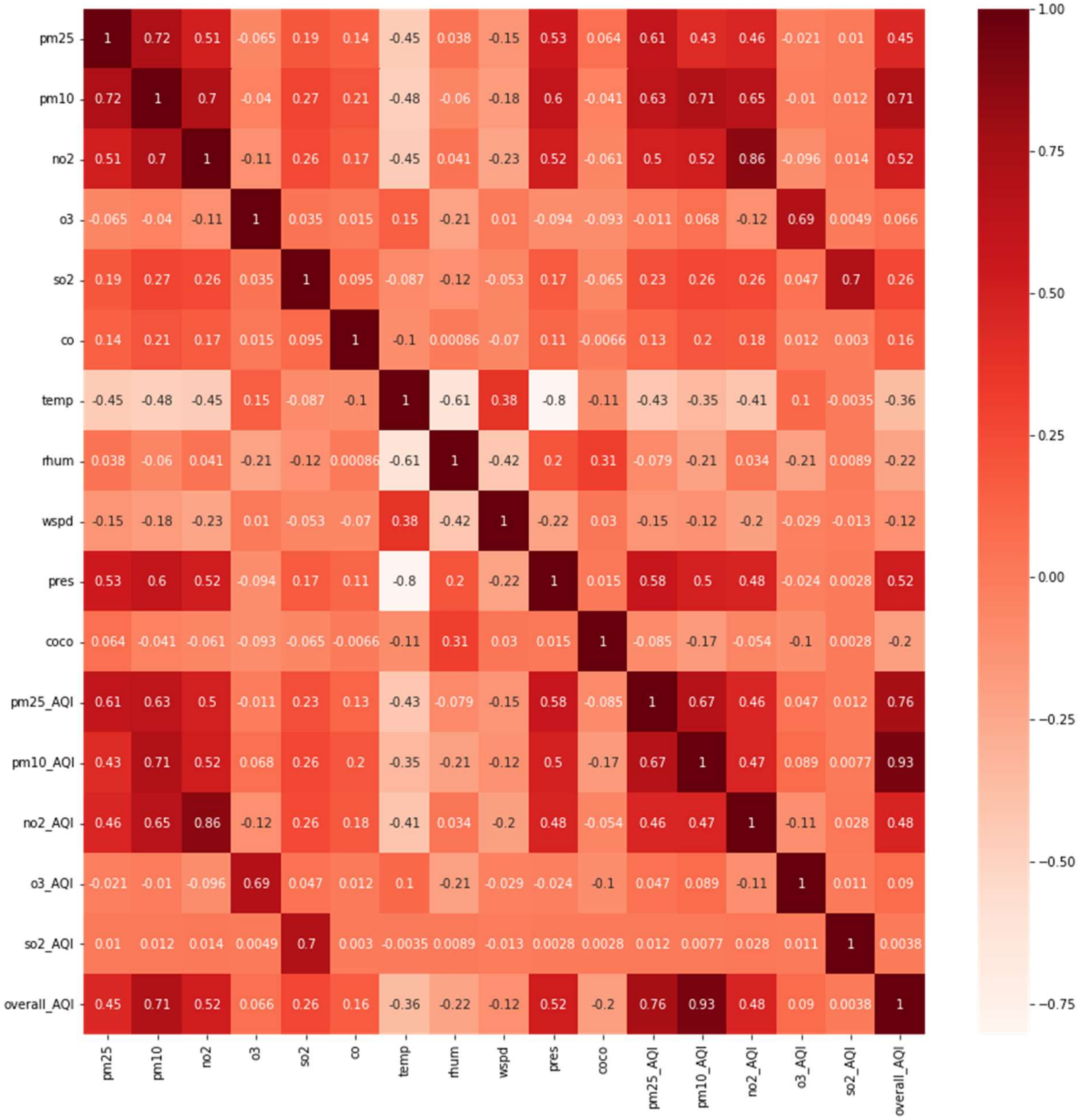


Fig 3: Spacial temporal correlation analysis

2.4 Time Series Data Construction.

In this experiment, the time of the input data is taken as a sequence, and the input data are segmented into two dimensions and constructed into three dimensions using a time series construction approach. This function takes in a sequence of data-points gathered at equal intervals, along with time series parameters such as length of the sequences/windows, spacing between two sequence/windows, etc., to produce batches of timeseries inputs and targets. Here use batch size as 4 and sequence length 12. In this experiment, training, validation, and test sets of the built time series data are created. For training, validation, and evaluation, the prediction model uses the first 47 data points from each layer as input and the AQI value from the 48th layer as output.

3 TRAINING AND TESTING THE MODEL

The time series data undergoes training with several deep learning models such us long shot term memory, , bidirectional gated recurrent unit, gated recurrent unit, bidirectional long short term memory, convolutional LSTM and convolutional bidirectional LSTM. All models are trained separately, and these models are saved. The proposed models are tested with testing data and predict PM2.5, AQI and evaluate the predicted vs actual values with evaluation metrics. The accuracy of proposed CNN BI-LSTM is much higher than other models in testing.

Two hidden layers are utilised in the GRU, LSTM, BI-GRU, and BI-LSTM networks, with two hundred units in the layer one and hundred on next . Network's final layers output is connected to a denc layer with just one end neuron. There is a 40% dropouts applied in between the layers. For the hidden layers , the LeakyReLU is used as activation function.

Each network in CNN-LSTM and CNN-Bi-LSTM contains a 1D CNN, which has a convolutional layer with 128 feature detectors sequentially and a convolution window length of 2 with causal padding. Afterwards, there is a MaxPooling layer of one dimensional. This is related to recurrent network employs a dense layer with one output neuron and a 50% dropout rate. Adam was a learning rate optimizer that we utilised (0,001)

3.4 DEEP LEARNING MODELS

This study compares the PM2.5 prediction capabilities of several deep learning algorithms. The RNN architectures are long shot term memory, gated recurrent unit, bidirectional long short term memory, bidirectional GRU, convolutional LSTM and convolutional bidirectional LSTM. Subsequently, we will provide a summary of each network.

3.4.1 LSTM:

The LSTM network is a type of RNN in which the hidden neuron is replaced by a memory block that stores and retrieves data, allowing the system to model long-term dependencies. Each tiny unit receives three sigmoid functions from the LSTM. Input, forget, and output gates are what they are called. The gating function, which regulates the entrance and outflow of data, is implemented using these three functions.

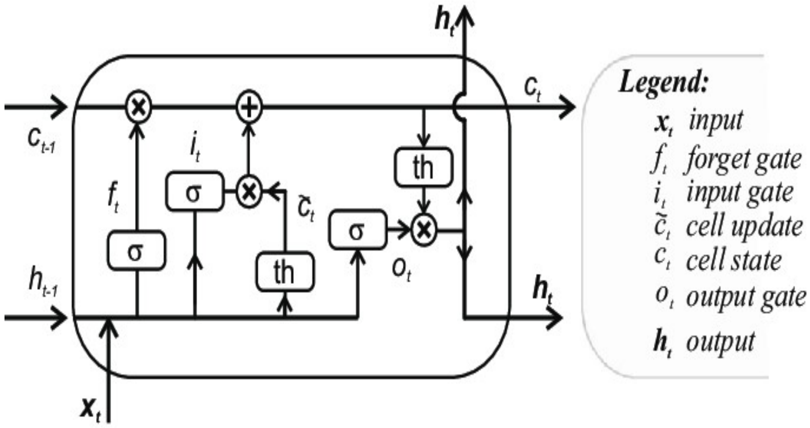


Fig 4: LSTM architecture

The forget gate's function is to decide which data in the memory cells should be deleted. It establishes the maximum amount of data that can be transferred from memory cell Ct1 to memory cell Ct at the current time. The input gate is the following gate. Its primary role is to control how much fresh input data can now reach memory cell Ct. The output gate is the third gate. Its primary job is to verify whether the data currently stored in memory cell Ct can be used to access the hidden layer state ht. The memory cell must then be updated via the LSTM architecture. According to the LSTM's construction, input gates and forget gates are required for updating memory cells. The tanh function is used to produce the candidate memory cell Ct, which is used to enter the input gate. How much old state information can enter the new memory cell is determined by multiplying the forget gate's output by the old state. The candidate memory cells are multiplied by the input gate's output to calculate the amount of information that has to be updated. To acquire memory cells that have recently been updated, they are added linearly. Finally, the model will produce the data. The data is standardised to the range [-1, 1] after being processed by the tanh function (tanh (Ct)). The quantity of data output to the hidden layer is then controlled by multiplying the output gate result by the normalised data.

3.4.2 GRU

The GRU architecture is an expansion of the LSTM architecture. GRU is an adaptation of LSTM that combines the i/p gate and forget gate into a single update gate. Additionally, some additional changes are made while combining the cell state with the concealed state. Based on LSTM, GRU made two significant improvements. The fewer gates are the first thing to mention. An update gate is created by GRU is by fusing the i/p gate and forget gate in the long short term memory. Reset gate is another. The reset gate's function is to regulate the memory space loss. Update gate comes first. What much of prior data can be passed on to the future is the update gate's primary role. The information of the previous moment and the current moment are linearly transformed by the right multiplication weight matrix. The added data is sent to the updategate and multiplied by the sigmoid function. The resulting value is between [0, 1].

The update gate serves the same purpose as the output gate in the LSTM by deciding whether to update the hidden state to the new state.

The reset gate is the following gate. The reset gate's primary job is to establish how much past data cannot be transferred to the present. The information from the prior instant and the present moment are linearly converted, much like with the update gate's data processing. The weight matrix of these two data processing has various values and functions. The linear self-renewing memory unit is cancelled by GRU, which is the second distinction between LSTM and GRU. In the concealed unit, it immediately does linear self-updating utilising gating.

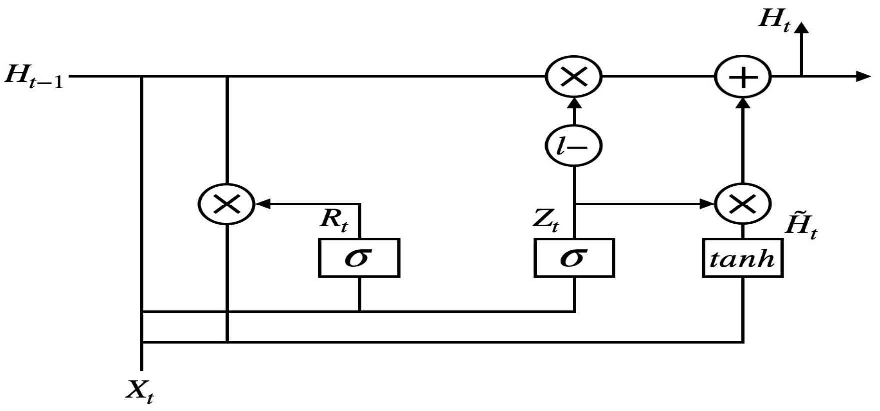


Fig 5: GRU architecture

GRU no longer stores memory data in discrete memory cells. Utilizing hidden units, it immediately records the historical condition. The reset gate regulates the data size of the current information and the memory information. The creation of fresh memory data keeps progressing. The memory data is entirely erased when the reset gate is at 0. On the other hand, if the reset gate is 1, then all of the memory data has been transmitted.

3.4.2 Bi-LSTM

When processing time, traditional RNN and LSTM usually ignore information about the future; nevertheless, Bidirectional-LSTM can able to exploit this data.. The forward and rear layers of each sequence are each made up of two LSTM networks, which is the fundamental concept behind the main structural principle of Bi-LSTM. In addition to this, each of the LSTM networks is coupled to one input layer as well as one output layer. Through this structure, the output layer is able to retrieve historical data pertaining to each point obtained from the input sequence as well as gather data pertaining to the points’ upcoming states.

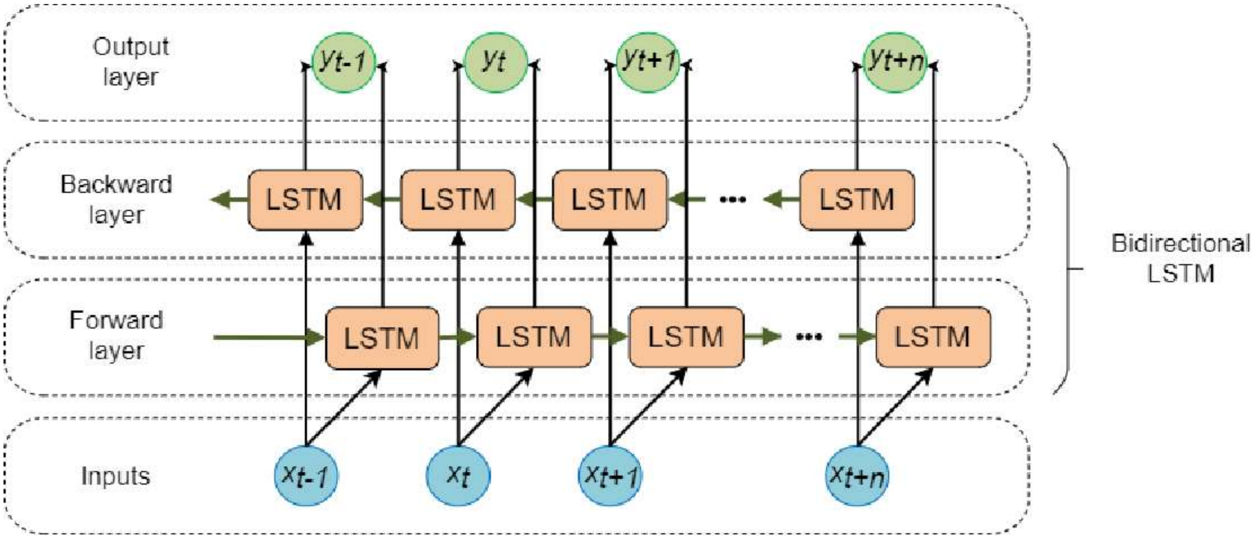


Fig 6:Bi-LSTM architecture

3.4.4 Bi-GRU

A bidirectional GRU or Using two GRUs, one of which receives input in a forward manner and the other in a backward way, the sequence processing model known as Bi-GRU is created. It is a bidirectional recurrent neural network with only the input and forget gates

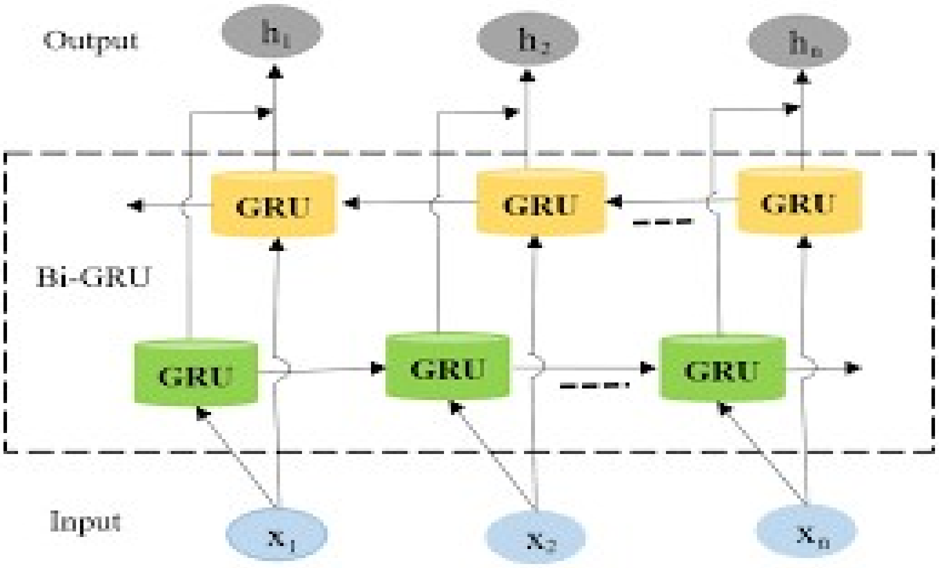


Fig 7: BI-GRU architecture

3.4.5 CNN-LSTM

Whether input networks consist of photos multidimensional tensors, the adoption of the standard CNN design is the optimum option. Convolutional LSTM, often known as ConvLSTM, is a novel variation of LSTM that was created since LSTM architectures are better suited for 1-D data. In this design, the output layer retains the LSTM cell rather than merely a 1-D vector since it incorporates a convolutional actions and input feature parameters. At each gate of a traditional LSTM, a convolution operation takes the role of matrix multiplication. We may argue that the ConvLSTM design combines CNN and long short term characteristics.. Combining LSTM and convolutional layers one block at a time is another method for processing spatiotemporal data. This architecture is known as Convolutional LSTM (CNN-LSTM). Convolutional layers extract key characteristics from the input data in the first portion of this model, and the outputs fattened in one dimensional can be utilised as information for the model's second component (long short term memory). Atlast, information must be moulded into the actual form of input information before travelling through final concealed layer.

3.4.6 CNN BI-LSTM

A CNN layer and a Bi-LSTM layer are both components of the hybrid CNN-Bi-LSTM model. It can use CNN to extract useful spatial information from all PM2.5-related parameters. Bi-LSTM is used to identify temporal characteristics in two directions of the hidden layer and resolve gradient disappearance and explosion issues in the context of time series. The suggested deep learning network primarily included embedding, 1D convolution, pooling, bilateral LTSM, dropout, flatten, and fully connected layers.

Two noteworthy aspects of the convolutional neural network (CNN), a feed-forward network, are local connection and parameter sharing. Local connection is the state of partial connectedness between two adjacent layers. The filter is used by CNN to implement parameter sharing (also called convolution kernel). The filter moves over the data and becomes tangled up with every piece of the data. Here, 1D convolution was utilized. Kernel mobility in a 1D CNN is unidirectional. Data for a 1D CNN's input and output are in two dimensions used often with time-series data.

The optimum option when the input matrix is a 2-D or 3-D tensor, such as an image or a video, is to employ the conventional CNN design. An updated variation Bi-long short term momory called Conv bidirectional LSTM has been developed since the framework of Bi-LSTM is better suited for 1-D Data. This architecture keeps a cell with a convolutional actions and an input feature parametres in the final layer as opposed to a simple one - dimensional vectors. At each gate of a traditional LSTM, a convolutional operations takes the role of matrix multiplication. Conv Bi-LSTM framework combines CNN and Bi-LSTM Network features, so to speak. Combining CNN and LSTM layers one block at a time is an additional strategy for handling spatiotemporal data. Convolutional Bidirectional LSTM is the name of this design Convolutional layers extract key characteristics from the input data in the first portion of this model, and the outputs fattened in one dimensional can be utilised as information for the model's second component (long short term memory). Atlast, information must be moulded into the actual form of input information before travelling through final concealed layer.

A different type of neural net framework than multiple layer perception is the recurrent neural network. The RNN shares weights and is especially well suited for sequence analysis tasks like linguistic translation and semantic interpretation. The previous hidden state, which was calculated by where was an activation function and was a bias, as well as the current input were both factors affecting the hidden state at the time interval. Where was a bias, an activation function, it was used to compute the output at the time interval. A catastrophic problem with the RNN, known as vanishing gradient, existed for lengthy sequences. The LSTM is one of the better methods for addressing the vanishing gradient.

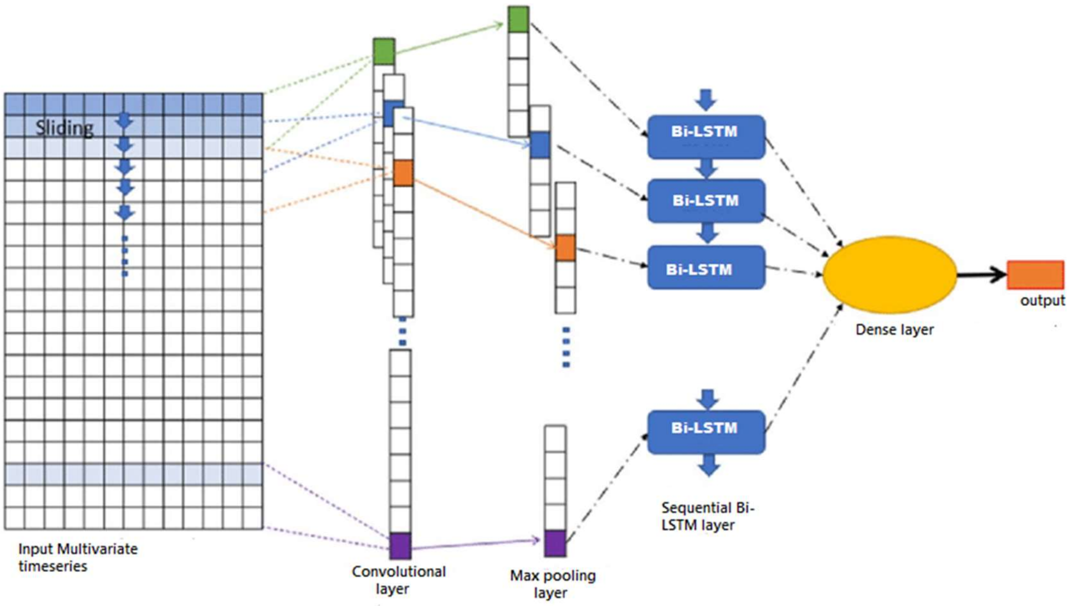


Fig 8: CNN BI-LSTM architecture

Two convolutional layers make up the one-dimensional (1D) CNN found in the CNN Bi-LSTM framework. A series of filter (or kernels) make up the convolution layers, and its parameters must be learnt throughout training. The filters are often smaller in size than the input images. The activation map is produced by each filter's convolution with the image. It is followed by a maximum pooling layer after background subtraction. The feature maps' sizes are decreased by pooling layers. As a result, it decreases the number of elements to learn and the amount of network processing required. A flatten layer is used to combine all of the 2-Dimensional arrays from the pooled feature maps into a single, lengthy continuous linear vector.

Two hidden layers with opposing orientations are linked to the same outcome via bidirectional RNN (BRNN). This kind of generative deep learning allows the output layer to concurrently receive data from previous (backwards) and future (ahead) states. then a thick layer made up of only one output neuron comes next. Each neurons in the dense layer of a neural network receives the input from all the neurons in its preceding layer due to the layer's dense connections. The neuron in the dense layer of a deep-connected neural network receives input from all the neurons in the layer below it.

3.5 SOFTWARE REQUIREMENTS AND SPECIFICATION

The software used for the project:

1 Python

2 Google Collaboratory

1. PYTHON

Guido Rossum designed the object-oriented programming language Python in 1989. For quickly prototyping complex tasks, it is perfectly suited. In comparison to C or C++, it is protracted and includes interfaces to a number of OS system functions and libraries. NASA, BitTorrent,Google, YouTube, and other large companies all employ the Python programming language. AI, NLG, neural networks, and other cutting-edge computer science topics all make extensive use of Python programming. The Python Software Foundation is currently in charge of the open-source, high-level programming language that Guido van Rossum created in the late 1980s. It originated with the ABC language, which he co-created in the beginning of his professional life.Python is a powerful language that you can use develop games, write GUIs, and develop web applications. It's a high-level language. Reading and writing codes in Python is far like reading and writing regular English statements. As a result, they're not written in the machine-readable language, Python programs got to be processed before machines can run them. Python is an understood language. This implies that each time a program is run, its interpreter runs through the code and interprets it into machine-readable byte code. Python is an object-oriented language control users to manage and management data structures or objects to make and run programs. The quality of Python is excellent throughout. In Python, all classes, data types, functions, and methods are treated equally. Programming languages are developed to meet the needs of users and programmers for an effective tool to construct programmes that have an influence on people's lives, way of life, economy, and society. By boosting productivity, improving communication, and boosting power, they help improve life. When a language falls short of expectations, it dies and is replaced by a more powerful language, which makes it obsolete. Python is an artificial language used for programming that has endured and persisted over time in a variety of enterprises, sectors, and among programmers and ordinary users. For people who wish to get started with coding and discover more about it, it is a live, vibrant, and really beneficial language.

- Interpretation of Python: The interpreter processes Python while it is being used. You may run your application right away without compiling it beforehand. This is comparable to PERL and PHP . Python is Interactive: You can really sit at a Python prompt and communicate with the interpreter directly to create your programmes.
- Python is object-oriented: Python supports the object-oriented style or approach of programming, which wraps code into objects.
- Python is a Beginner's Language: Python is a fantastic language for beginning programmers and facilitates the building of a broad range of applications, from basic text processing to web browsers .
- A simple and clear language that is equally as potent as that of the main rival
- Everyone may contribute to its development because it is open source. Code that is as easy to comprehend as simple English

2. GOOGLE COLABORATORY

For anyone who needs them to develop a machine learning or deep learning model, Google Colab offers free access to GPUs and TPUs. A better version of Jupyter Notebook is what Google Colab is. An programme called Jupyter Notebook enables editing and execution of Notebook documents using a web browser or an Integrated Development Environment (IDE). It will operate using Notebooks in place of files. Text, photos, figures, tables, graphs, equations, and a variety of other graphical data can all be found in notebook papers, along with executable lines of code. Simply said, writing human-readable executable documents is what Notebook documents are. Cells serve as the foundation of a Notebook. Cells make up every element of a notebook.There are two types of :

• Text cell: Text, photos, links, and much more can all be included in text cells. Double-clicking a text cell will allow you to change what is within. Markdown is supported in the text cell.

• Code cell: - The executable code is contained in a code cell. The run button to the left of a code cell allows to run the cell's code. The output of a cell is shown underneath the cell when it is executed. The interesting features that each contemporary IDE offers are abundant in Google Colab, in addition to many others.

Below is a list of some of the more fascinating aspects.

• Interactive lessons for learning neural networks and machine learning.

• Create and run Python 3 programmes without a local setup.

• Use the Notebook to run terminal commands.

• Bring in data from outside sources

•To Google Drive, save notebooks

.• Google Drive notebook import

t• No-cost cloud computing, GPUs, and TPUs

.• Integrate with PyTorch, Tensor Flow, Open CV

.• Import or publish directly from/to GitHub.

CHAPTER 4

RESULTS AND DISCUSSIONS

The prediction of PM2.5 concentrations was simulated in this work using multiple deep-learning methods. The historical observation PM2.5 data are compared with the calculated PM2.5 from long shot term memory, gated recurrent unit, bidirectional long short term memory, bidirectional gated recurrent unit, convolutional long short term memory and convolutional bidirectional long short term memory evaluated in one lag day in this part.

Each network makes an attempt to anticipate the outcomes as precisely as possible. In this study, we used Mean Squared Error as a cost function for all networks. For neural networks, an epoch is the number of iteration steps; during each epoch, the network simulates the streamflow time series once. Recurrent networks allow for the arbitrary selection of neurons or network layers. All of the recurrent network models in our study have the same architecture in order to compare them with one another.

Two hidden layers are utilised in the GRU, LSTM, BI-GRU, and BI-LSTM networks, with two hundred units in the layer one and hundred on next . One output neuron-only dense layer is linked to the output of the network's final layer. There is a 40% dropouts applied in between the layers. For the hidden layers , the LeakyReLU is used as activation function.

Each network in CNN-LSTM and CNN-Bi-LSTM contains a 1D CNN, which has a convolutional layer with 128 feature detectors sequentially and a convolution window length of 2 with causal padding. Afterwards, there is a MaxPooling layer of one dimensional. This is related to recurrent network ,it employs a thick layer with an output data neuron and applies a drop out of 50% between layers.. Adam was a learning rate optimizer that we utilised (0,001)

4.1EVALUATION INDEX OF THE MODELS

In this article, three measures are used to assess the model's efficiency: mean absolute error , mean squared error , and coefficient of determination

MAE

The true prediction error scenario can be better described by the arithmetic mean of the absolute variances among the reality value and the predicted data of all samples, known as Mean Absolute Error.The following is the calculating formula:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \qquad (5)$$

RMSE

The square-root of the mean square of all mistakes is known as RMSE (Root Mean Square Error). It could accurately reflect the forecast error's accuracy. The following is the calculating formula:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}(y_i - \hat{y}_i)^2} \qquad (6)$$

MSE

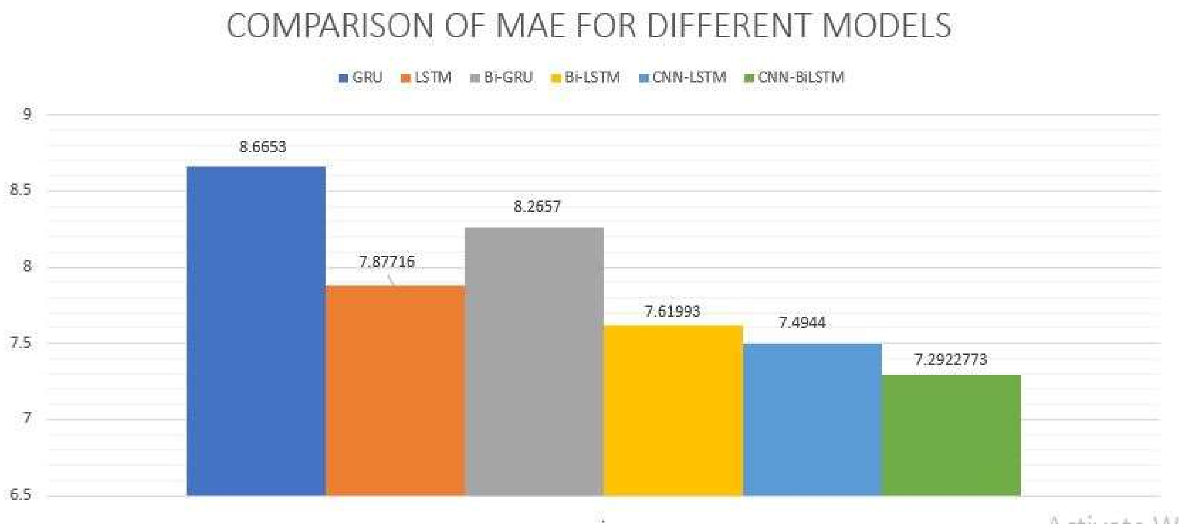
An estimator's mean squared error calculates the sum of the squared difference between the predicted values and the actual value (MSE). MSE, which corresponds to the expected value of the squared error loss, is a risk function. The following is the calculating formula:

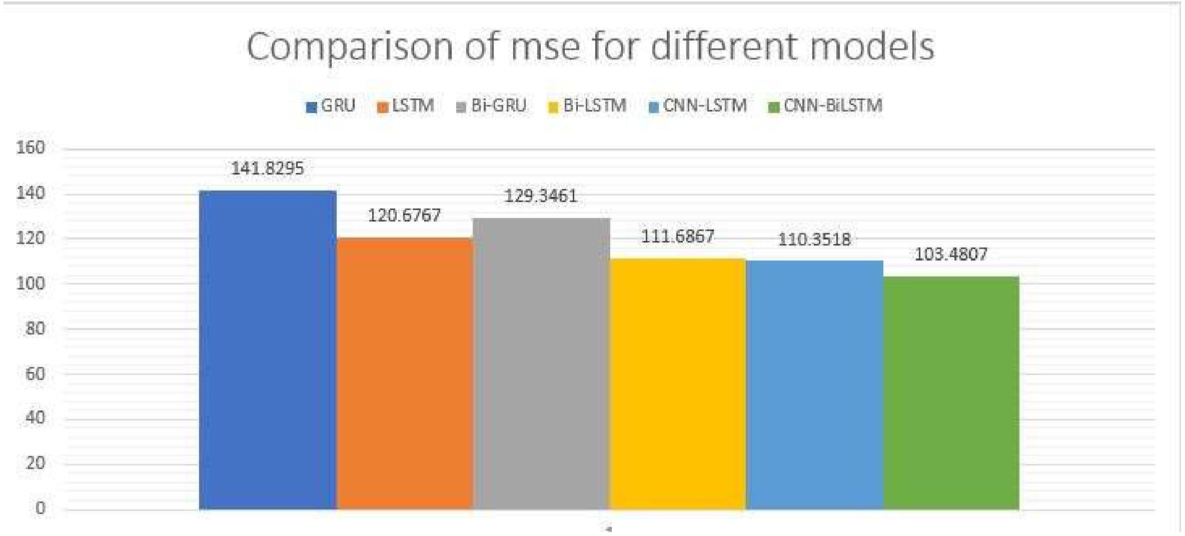
$$MS \quad = \frac{1}{n}\sum_{i=1}^{n}(y - \hat{y}_i)^2 \qquad (7)$$

4.2 TRAINING AND VALIDATION RESULTS

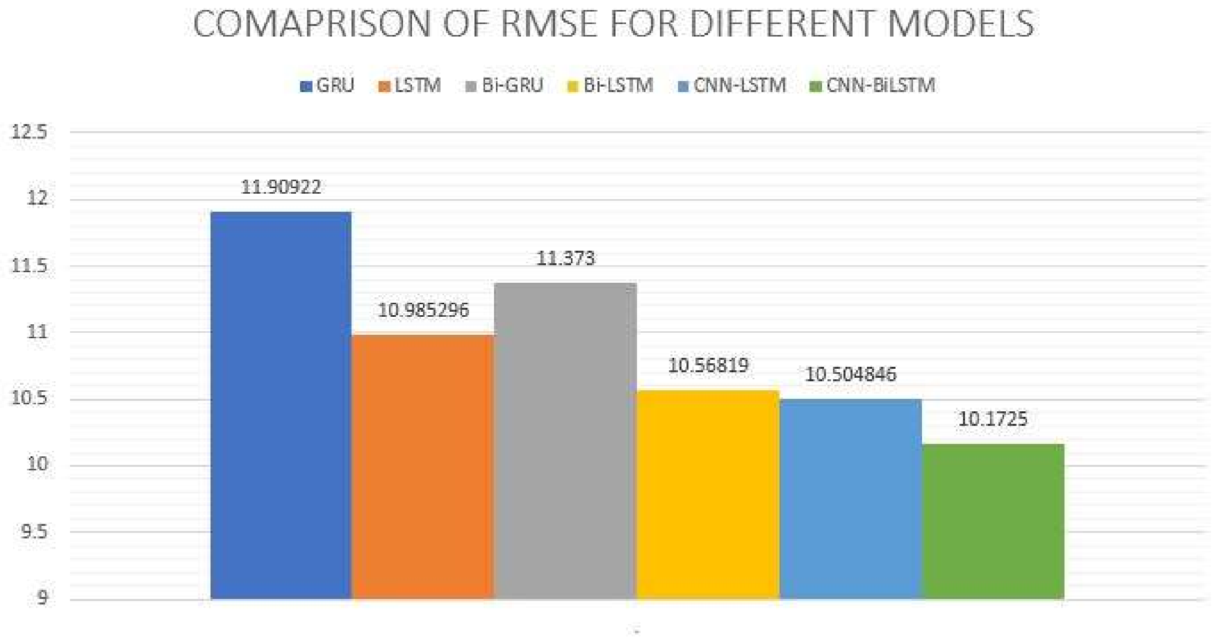
Model Comparison

The MAE, RMSE, and MSE for six models are shown in Figures 4.1, 4.2, and 4.3. These numbers fall within the range of PM2.5 concentrations that are expected and actual.





-



The MAE, RMSE, and MSE values produced by the model prediction models for the concentration of PM2.5 in the air are listed in Table 2. The CNN BILSTM models' RMSE values were the lowest of all the models. The CNN Bi-LSTM fared better than other models in comparison to the identical settings, according to the results.

MODELS	MAE	MSE	RMSE
GRU	**8.66531**	**141.8295**	**11.90922**
LSTM	**7.87716**	**120.6767**	**10.985296**
BI-GRU	**8.26573**	**129.3461**	**11.3730**
BI-LSTM	**7.61993**	**111.68669**	**10.56819**
CNN LSTM	**7.49447**	**110.35179**	**10.504846**
CNN BILSTM	**7.29228**	**103.480705**	**10.1725**

The MAE,MSE, and RMSE of BI-LSTM are reduced when LSTM is contrasted with Bi-LSTM. Therefore, BI-LSTM was superior than LSTM: MAE decreased from 7.87716 to 7.61993, MSE from 120.6767 to 111.68669, and RMSE decreased from 10.9853 to 10.5682.

• Based on assessment criteria, we may conclude that CNN LSTM outperforms BI-LSTM since MAE lowers from 7.61993 to 7.49447, MSE drops from 111.68669 to 110.3518, and RMSE drops from 10.5682 to 10.5049.

• The suggested model in this research has the least MAE, RMSE, and MSE values when compared to CNN LSTM and CNN BI-LSTM. the MSE falls from 110.3518 to 103.4807, the MAE from 7.4945 to 7.2923, and the RMSE from 10.50485 to 10.1725.

Overall, Table 1's data demonstrate that among the six models, CNN BI-LSTM performs the best. The MAE is 7.2923 and the RMSE is 10.1725, which is the least of compared models and has the best predicting respectively.

The comparison between the predicted findings and the actual values for PM2.5 data is presented below based on our suggested prediction approach and another comparable model. The numbers show that the suggested strategy is able to produce prediction results that are much closer to the actual values than other models.

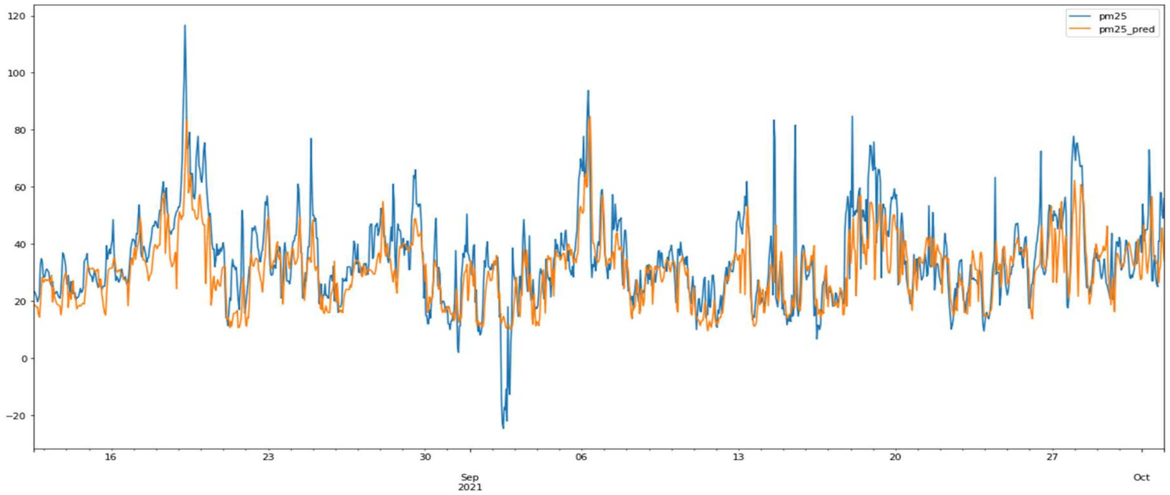


Fig 12: Comparison between the prediction results and the actual values in lstm

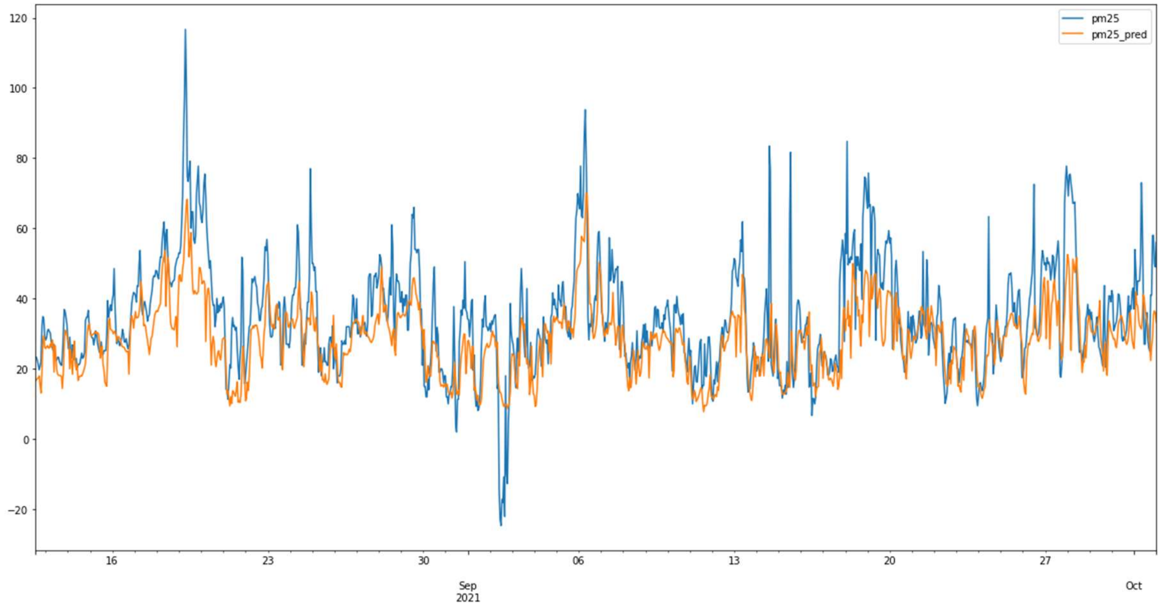


Fig 13: Comparison between the prediction results and the actual values in GRU

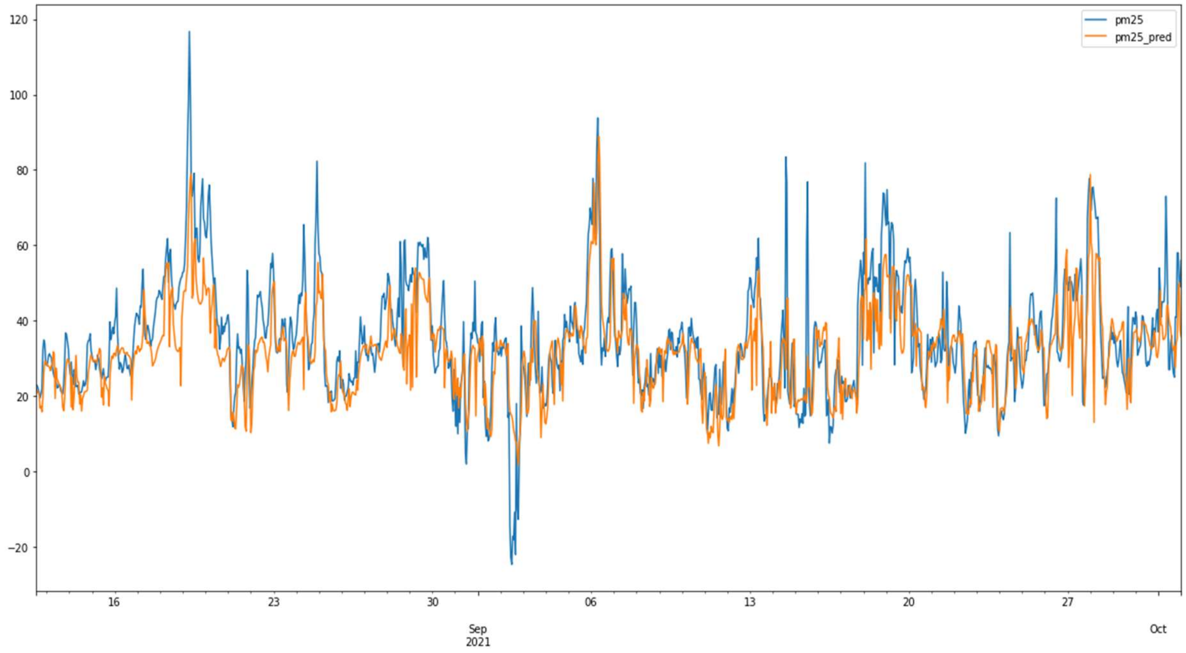


Fig 14: Comparison between the prediction results and the actual values in BILSTM

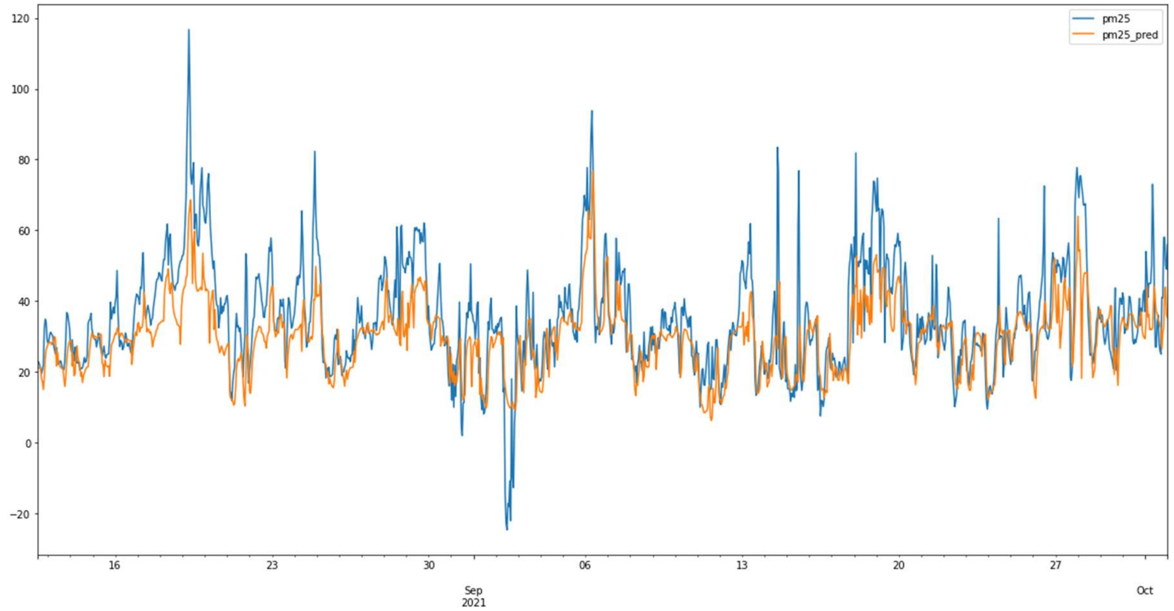


Fig 15: Comparison between the prediction results and the actual values in BI GRU

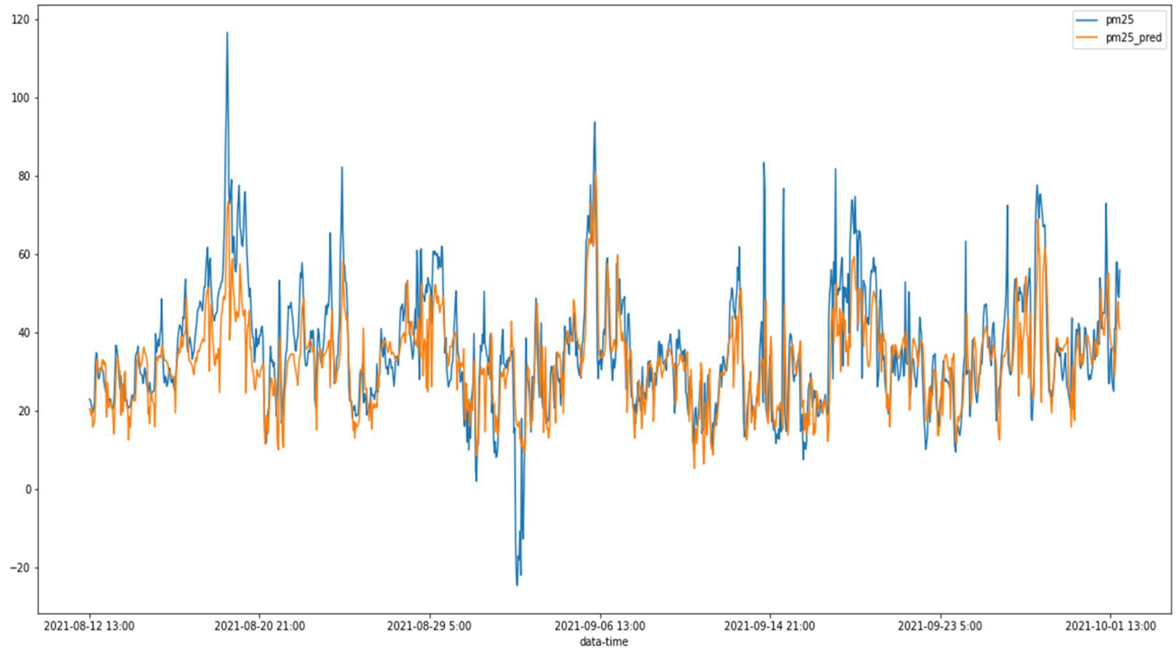


Fig 16: Comparison between the prediction results and the actual values in CNN LSTM

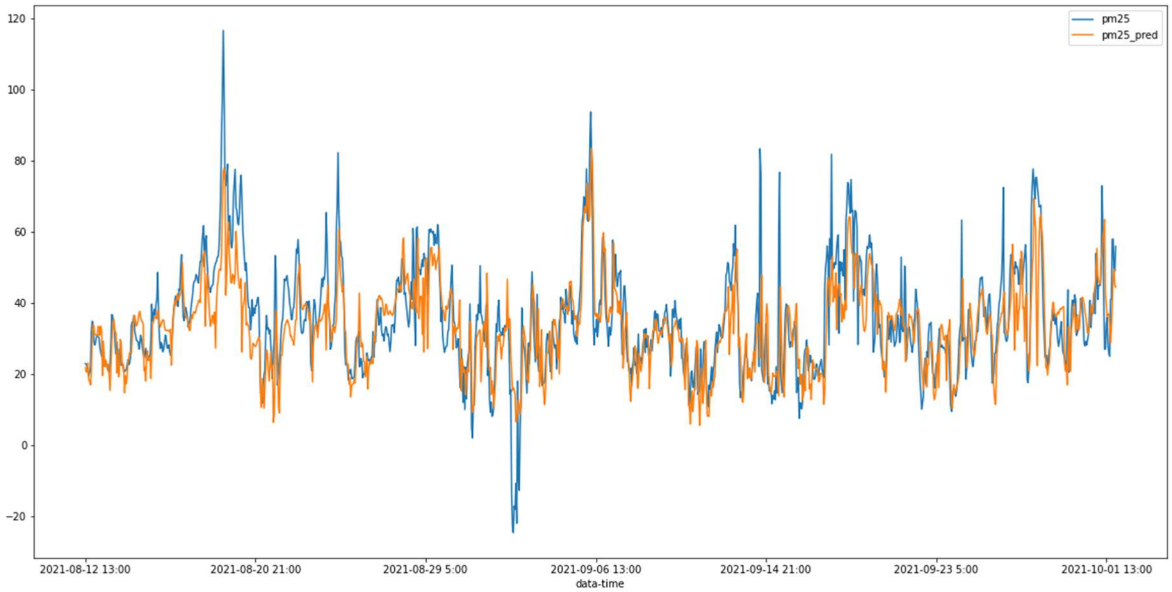


Fig 17: Comparison between the prediction results and the actual values in CNN BILSTM

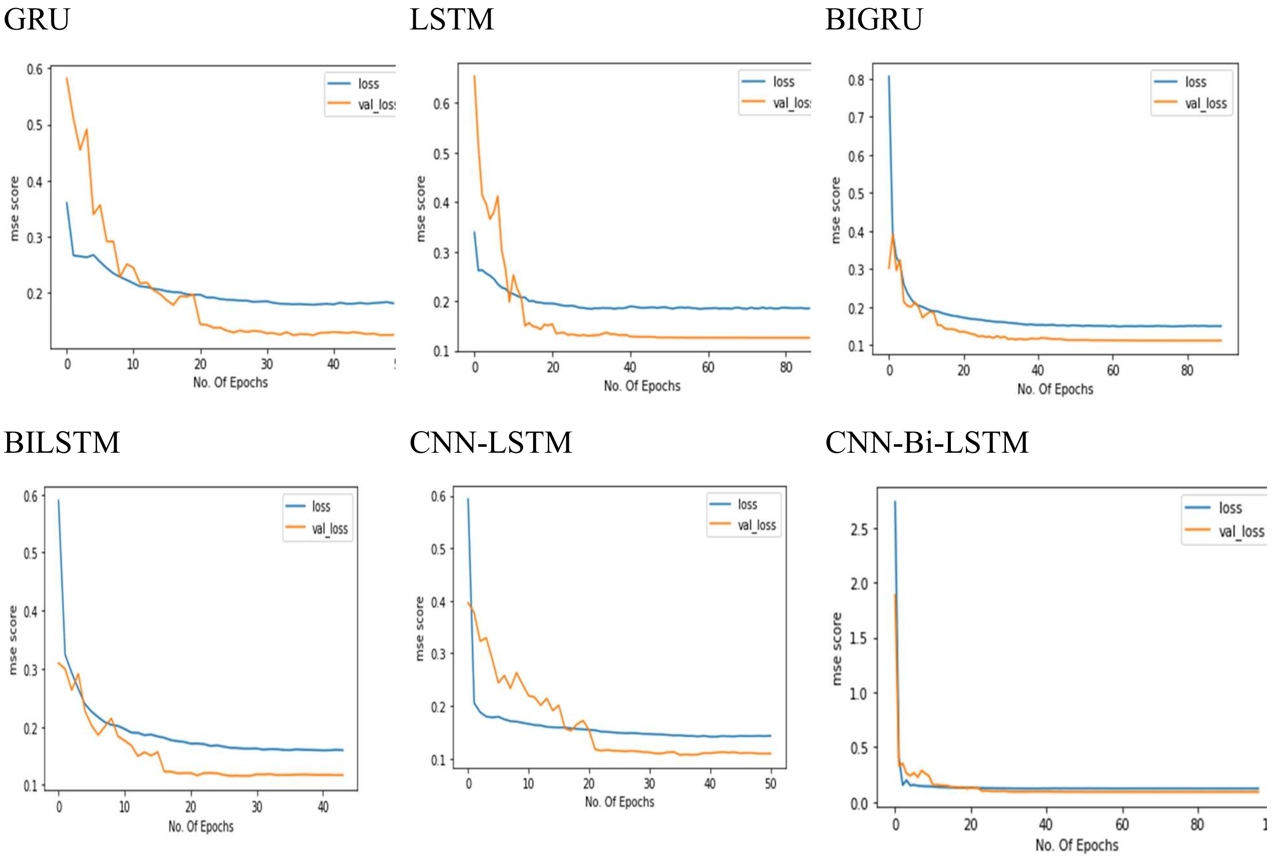
Model convergence.

Checking if CNN-Bi-LSTM often converges during training is important once the model has been constructed and its parameters have been established. In this experiment, the 1-0 loss function is used to demonstrate its convergence when all parameters are the same and the epoch is 400. The convergences of the CNN-Bi-LSTM in this experiment are depicted in Fig. below. CNN-BI-convergence LSTM's speed in this experiment is quicker than CNN-due LSTM's to CNN-BI-loss LSTM's function being smaller than CNN- LSTM's.

A metric called the training loss quantifies how well a deep learning model fits the traning data. It assesses the error of the model on the training set. The training set is a portion of the dataset that was first used to train the model, it should be emphasized. By summing the total number of errors for each sample in the training set, the training loss is determined. The success of a deep training procedure on the cross validation is measured using a statistic called validation loss, on the other hand. A portion of the dataset designated to assess the effectiveness of the model is the validation data. The errors for each example are added to determine the validation loss, same as the training loss.

MODELS	LOSS	VALIDATION LOSS	TIME PER EPOCH (S)
CNN BI-LSTM	**0.1430**	**0.1092**	**36 s**
CNN LSTM	**0.1551**	**0.1228**	**46 s**
LSTM	**0.1853**	**0.1260**	**42 s**
GRU	**0.1808**	**0.1267**	**42 s**
BI-LSTM	**0.1501**	**0.1123**	**80 s**
BI-GRU	**0.1598**	**0.1170**	**82 s**

Table 3: Training loss and validation loss in deep learning models


GRU

LSTM

BIGRU

BILSTM

CNN-LSTM

CNN-Bi-LSTM

Fig 18: Graphical representation for Training loss and validation loss in models

As shown in Figure 18. It can be seen that when the epoch is almost 10, the loss value of the CNN-BiLSTM is 0.1430 which is he smallest among models and the training time taken is 36 seconds per epochs for CNN BI-LSTM which is fastest while comparing other models.

CHAPTER 5
CONCLUSION

The suggested approach, which uses historical and meteorological data to predict air pollution, produces notable results. In this study, a CNN-Bi-LSTM-based air pollution prediction model is presented. The CNN Bi-LSTM is the deep learning model with the greatest overall assessment of prediction outcomes when compared to gated recurrent unit, long short term memory, bidirectional LSTM, CNN LSTM, and bidirectional GRU. First, a correlation analysis was performed on the stations' historical data. A characteristic with a stronger correlation coefficient with the PM2.5, meteorological information, and correlation with other stations was chosen after experimental comparison. Second, with the hybrid model that was presented, and employed convolutional neural network to successfully take out the spatial features of and the inner characteristics between various variables concurrently. Temporal characteristics and a more precise and reliable prediction effect were obtained using Bi-LSTM. The training time of CNN Bi-LSTM is decreased by 12.5 and 50.9 percent when compared to Bi-LSTM and LSTM, respectively, and CNN Bi-LSTM has the best air quality prediction results. The experiment's findings indicate that the CNN Bi- LSTM's MAE and RMSE both reduce by 0.20224 and 0.33256, respectively. The key conclusions of this project, as determined through performance review and outcomes comparison, are as follows: This model has great accuracy and stability and can efficiently recover the spatially and temporally aspects of the data using CNN and Bi-LSTM.

5.1 FUTURE ENHANCEMENT

This study introduces a hybrid approach that makes use of deep learning algorithms to assess and forecast the air quality index. In comparison to other models, CNN BI-LSTM provides more accuracy. The focus of future work will be on optimising the architectures and additional deep learning models. Various problems and study directions may be taken into consideration for future research. Future studies should take into account traffic factors when forecasting air pollution since these elements have an effect on the growth in air pollution concentration both directly and indirectly. Future predictions might be more accurate if the missing characteristics are handled and the anomalies in the pollution dataset are predicted. Also we can achieve better results on grid-based predictions by implementing advanced interpolation models on the ground-based PM2.5 sensor input data.

REFERENCES

1. Garg S, Jindal H. ***Evaluation of time series forecasting models for estimation of pm2.5 levels in air***, 2021 6th international conference for convergence in technology (I2CT). IEEE; 2021
2. Xayasouk T, Lee H, Lee G. ***Air pollution prediction using long short-term memory (lstm) and deep autoencoder (dae) models***. Sustainability, 2020
3. Jeya S, Sankari L. ***Air pollution prediction by deep learning model.***, 2020 4th international conference on intelligent computing and control systems (ICICCS) IEEE; 2020
4. Zhao J, Deng F, Cai Y, Chen J. ***Long short-term memory-fully connected (lstm-fc) neural network for pm2.5 concentration prediction***. Chemosphere. 2019
5. Rijal N, Gutta RT, Cao T, Lin J, Bo Q, Zhang J. ***Ensemble of deep neural networks for estimating particulate matter from images***, 2018 IEEE 3rd international conference on image, vision and computing (ICIVC). IEEE; 2018.
6. Zhang L, Li D, Guo Q, ***Deep learning from spatio-temporal data using orthogonal regularizaion residual cnn for air prediction***. IEEE Access, 2020
7. Li J, Jin M, Li H. ***Exploring spatial infuence of remotely sensed pm2.5 concentration using a developed deep convolutional neural network model***. Int J Environ Res Public Health, 2019
8. Ma, J. ***, Improving air quality `prediction accuracy at larger temporal resolutions using deep learning and transfer learning techniques‟,*** Atmospheric Environment, 214(July),
   2019
9. Al-Janabi, S., Mohammad, M., & Al-Sultan, A. ***A new method for prediction of air pollution based on intelligent computation,*** Soft Computing, 2019
10. Ibrahim Kok,Suat Ozemir, ***A deep learning model for air quality prediction in smart cities"*** 2017 IEEE International Conference on Big Data (Big Data),2017

11. Chang, Y. S. et al. ***An LSTM-based aggregated model for air pollution forecasting,*** *Atmospheric Pollution Research*, 11(8), .2020

12. Bui, T. C., Le, V. D. and Cha, S. K.et.al. .***A Deep Learning Approach for Forecasting Air Pollution in South Korea Using LSTM***, Environmanental science,Arvix Learning ,2018

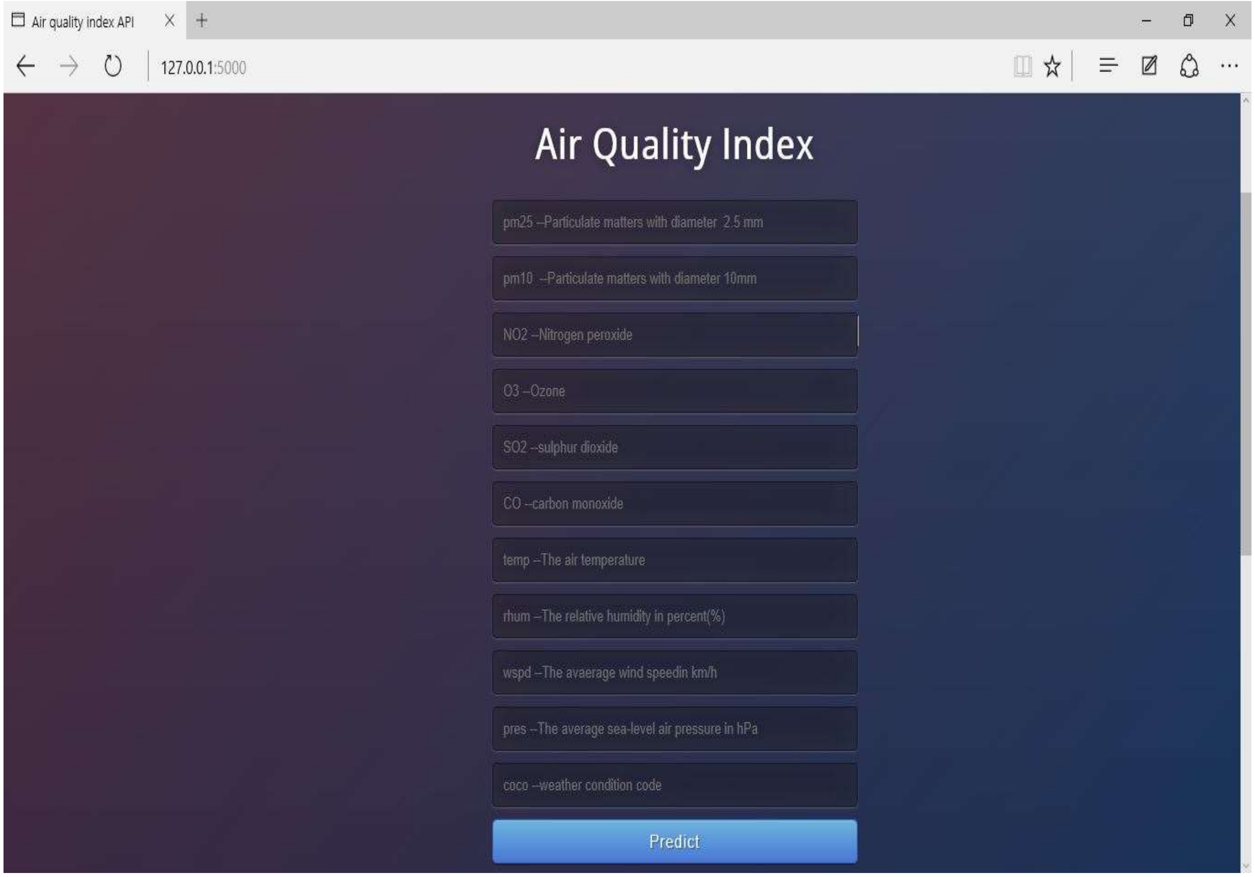
13. Chang, Y. S. et al. (2020) ***An LSTM-based aggregated model for air pollution forecasting***, Atmospheric Pollution Research, .apr.2020.

14. Le, V. D., Bui, T. C. and Cha, S. K. ***Spatiotemporal deep learning model for citywide air pollution interpolation and prediction,,***Proceedings - 2020 IEEE International Conference on Big Data and Smart Computing, BigComp 2020

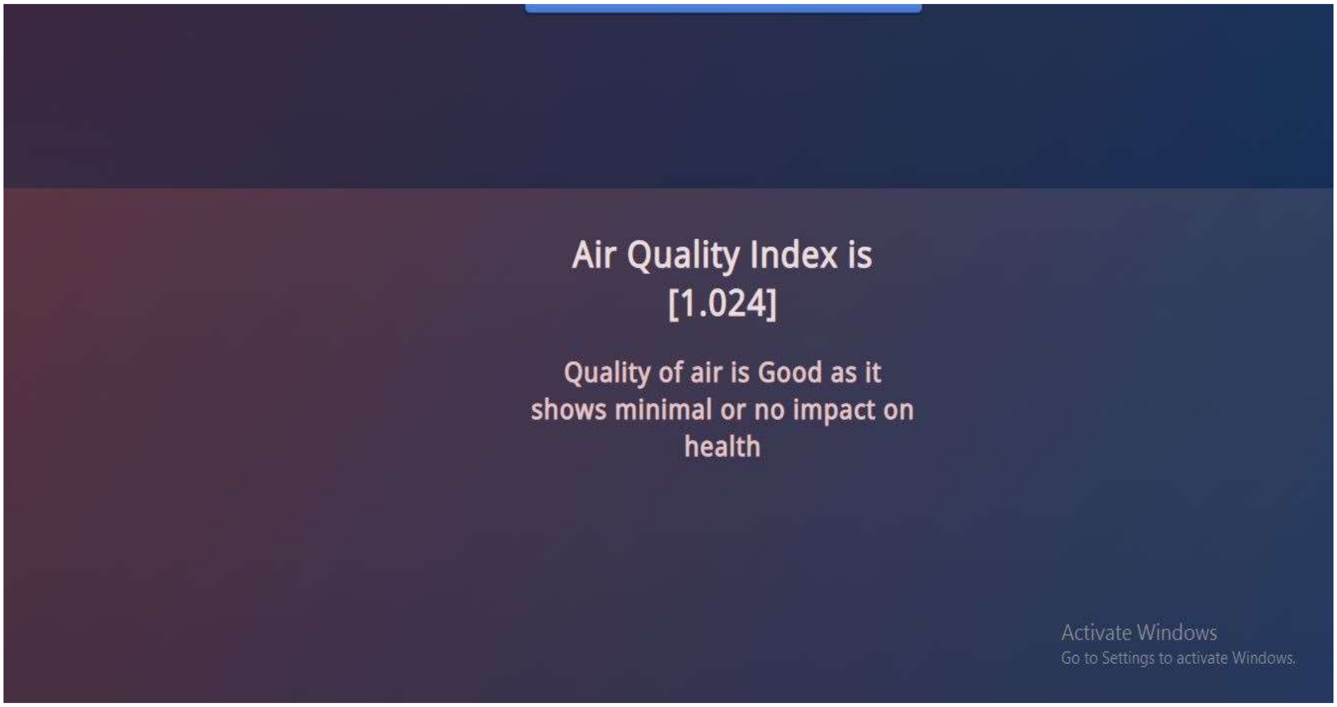
15. Sharma, E. ***Deep Air Quality Forecasts: Suspended Particulate Matter Modeling with Convolutional Neural and Long Short-Term Memory Network***s, IEEE Access, 2020

16. Abdellatif Bekkar , Badr Hssina ***Air-pollution prediction using deep learning approach***,Journal of Big Data, Springer, 2021

APPENDIX


Air quality index API
127.0.0.1:5000
Air Quality Index
pm25 --Particulate matters with diameter 2.5 mm
pm10 --Particulate matters with diameter 10mm
NO2 --Nitrogen peroxide
O3 --Ozone
SO2 --sulphur dioxide
CO --carbon monoxide
temp --The air temperature
rhum --The relative humidity in percent(%)
wspd --The avaerage wind speedin km/h
pres --The average sea-level air pressure in hPa
coco --weather condition code
Predict

Air quality index prediction input page

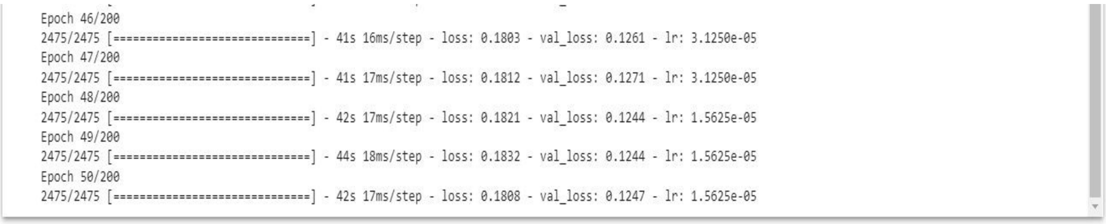


Air Quality Index is
[1.024]

Quality of air is Good as it shows minimal or no impact on health

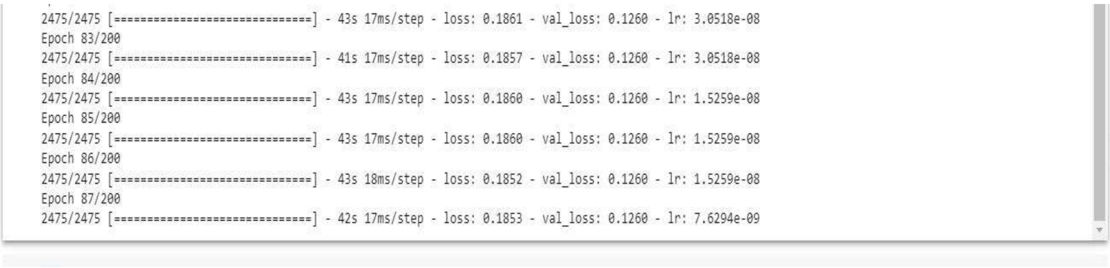
Predicted output

GRU



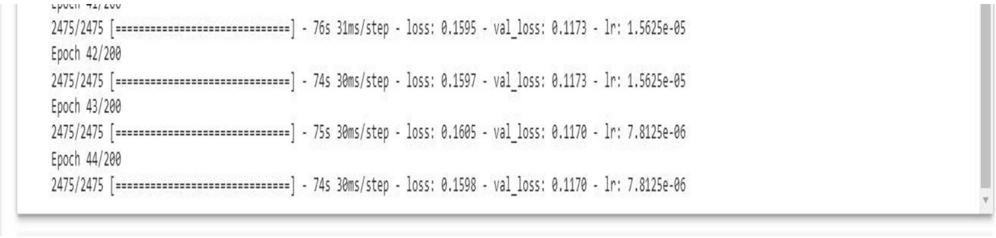
```
Epoch 46/200
2475/2475 [==============================] - 41s 16ms/step - loss: 0.1803 - val_loss: 0.1261 - lr: 3.1250e-05
Epoch 47/200
2475/2475 [==============================] - 41s 17ms/step - loss: 0.1812 - val_loss: 0.1271 - lr: 3.1250e-05
Epoch 48/200
2475/2475 [==============================] - 42s 17ms/step - loss: 0.1821 - val_loss: 0.1244 - lr: 1.5625e-05
Epoch 49/200
2475/2475 [==============================] - 44s 18ms/step - loss: 0.1832 - val_loss: 0.1244 - lr: 1.5625e-05
Epoch 50/200
2475/2475 [==============================] - 42s 17ms/step - loss: 0.1808 - val_loss: 0.1247 - lr: 1.5625e-05
```

LSTM



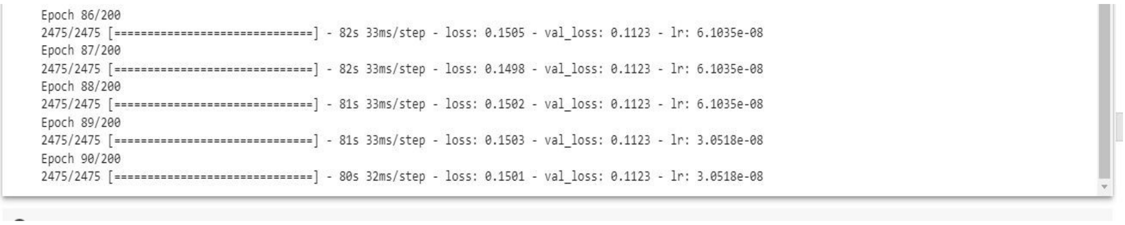
```
2475/2475 [==============================] - 43s 17ms/step - loss: 0.1861 - val_loss: 0.1260 - lr: 3.0518e-08
Epoch 83/200
2475/2475 [==============================] - 41s 17ms/step - loss: 0.1857 - val_loss: 0.1260 - lr: 3.0518e-08
Epoch 84/200
2475/2475 [==============================] - 43s 17ms/step - loss: 0.1860 - val_loss: 0.1260 - lr: 1.5259e-08
Epoch 85/200
2475/2475 [==============================] - 43s 17ms/step - loss: 0.1860 - val_loss: 0.1260 - lr: 1.5259e-08
Epoch 86/200
2475/2475 [==============================] - 43s 18ms/step - loss: 0.1852 - val_loss: 0.1260 - lr: 1.5259e-08
Epoch 87/200
2475/2475 [==============================] - 42s 17ms/step - loss: 0.1853 - val_loss: 0.1260 - lr: 7.6294e-09
```

BI-GRU



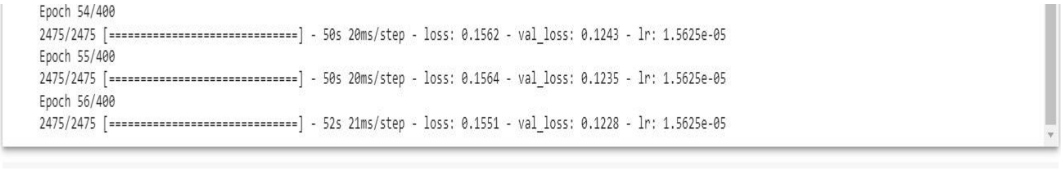
```
Epoch 41/200
2475/2475 [==============================] - 76s 31ms/step - loss: 0.1595 - val_loss: 0.1173 - lr: 1.5625e-05
Epoch 42/200
2475/2475 [==============================] - 74s 30ms/step - loss: 0.1597 - val_loss: 0.1173 - lr: 1.5625e-05
Epoch 43/200
2475/2475 [==============================] - 75s 30ms/step - loss: 0.1605 - val_loss: 0.1170 - lr: 7.8125e-06
Epoch 44/200
2475/2475 [==============================] - 74s 30ms/step - loss: 0.1598 - val_loss: 0.1170 - lr: 7.8125e-06
```

BI-LSTM



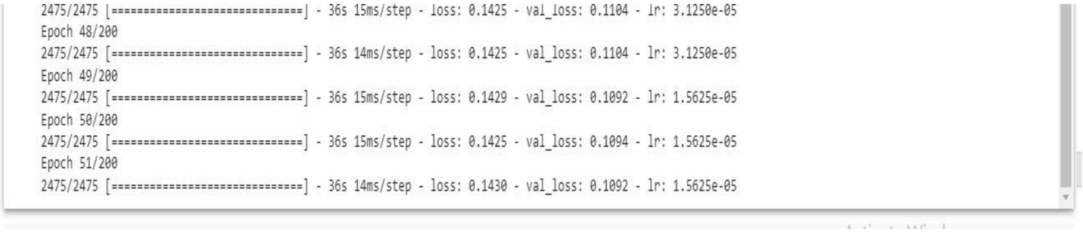
```
Epoch 86/200
2475/2475 [==============================] - 82s 33ms/step - loss: 0.1505 - val_loss: 0.1123 - lr: 6.1035e-08
Epoch 87/200
2475/2475 [==============================] - 82s 33ms/step - loss: 0.1498 - val_loss: 0.1123 - lr: 6.1035e-08
Epoch 88/200
2475/2475 [==============================] - 81s 33ms/step - loss: 0.1502 - val_loss: 0.1123 - lr: 6.1035e-08
Epoch 89/200
2475/2475 [==============================] - 81s 33ms/step - loss: 0.1503 - val_loss: 0.1123 - lr: 3.0518e-08
Epoch 90/200
2475/2475 [==============================] - 80s 32ms/step - loss: 0.1501 - val_loss: 0.1123 - lr: 3.0518e-08
```

CNN LSTM



```
Epoch 54/400
2475/2475 [==============================] - 50s 20ms/step - loss: 0.1562 - val_loss: 0.1243 - lr: 1.5625e-05
Epoch 55/400
2475/2475 [==============================] - 50s 20ms/step - loss: 0.1564 - val_loss: 0.1235 - lr: 1.5625e-05
Epoch 56/400
2475/2475 [==============================] - 52s 21ms/step - loss: 0.1551 - val_loss: 0.1228 - lr: 1.5625e-05
```

CNN BI-LSTM



```
2475/2475 [==============================] - 36s 15ms/step - loss: 0.1425 - val_loss: 0.1104 - lr: 3.1250e-05
Epoch 48/200
2475/2475 [==============================] - 36s 14ms/step - loss: 0.1425 - val_loss: 0.1104 - lr: 3.1250e-05
Epoch 49/200
2475/2475 [==============================] - 36s 15ms/step - loss: 0.1429 - val_loss: 0.1092 - lr: 1.5625e-05
Epoch 50/200
2475/2475 [==============================] - 36s 15ms/step - loss: 0.1425 - val_loss: 0.1094 - lr: 1.5625e-05
Epoch 51/200
2475/2475 [==============================] - 36s 14ms/step - loss: 0.1430 - val_loss: 0.1092 - lr: 1.5625e-05
```