

Course Project Report

Air Quality Index Prediction

Submitted By

Prathipati Jayanth - 211AI027

Balla Pradeep - 211AI009

as part of the requirements of the course

Data Science (IT258) [Feb - Jun 2023]

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

Bachelor of Technology in Artificial Intelligence

under the guidance of

Dr. Sowmya Kamath S, Dept of IT, NITK Surathkal

undergone at



DEPARTMENT OF INFORMATION TECHNOLOGY

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA, SURATHKAL

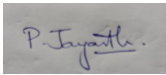
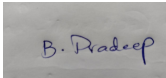
FEB-JUN 2023

DEPARTMENT OF INFORMATION TECHNOLOGY
National Institute of Technology Karnataka, Surathkal

C E R T I F I C A T E

This is to certify that the Course project Work Report entitled **“Air Quality Prediction”** is submitted by the group mentioned below -

Details of Project Group

Name of the Student	Register No.	Signature with Date
Prathipati Jayanth	2110504	 29/06/2023
Balla Pradeep	2110160	 29/06/2023

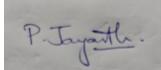
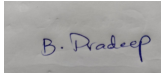
this report is a record of the work carried out by them as part of the course **Data Science (IT258)** during the semester **Feb - Jun 2023**. It is accepted as the Course Project Report submission in the partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Artificial Intelligence**.

(Name and Signature of Course Instructor)
Dr. Sowmya Kamath S

DECLARATION

We hereby declare that the project report entitled **“Air Quality Prediction”** submitted by us for the course **Data Science (IT258)** during the semester **Feb-Jun 2023**, as part of the partial course requirements for the award of the degree of Bachelor of Technology in Artificial Intelligence at NITK Surathkal is our original work. We declare that the project has not formed the basis for the award of any degree, associateship, fellowship or any other similar titles elsewhere.

Details of Project Group

Name of the Student	Register No.	Signature with Date
1. Prathipati Jayanth	2110504	 29/0/62023
2. Balla Pradeep	2110160	 29/06/2023

Place: NITK, Surathkal

Date: **29th June 2023**

Air Quality Prediction

Prathipati Jayanth¹, Balla Pradeep²

Abstract—Millions of people worldwide are impacted by air pollution, which is a serious health hazard. The Air Quality Index (AQI) is a vital instrument that offers insightful data on the state of the air quality right now and any potential negative consequences on health. This study employs two datasets. One is the Wuhan city dataset, which has climatic records from 2019/9/1 to 2020/12/29, and the other is the Shanghai city dataset, which has records from 2021/1/1 to 2022/4/23. PM2.5, PM10, SO2, NO2, O3, CO, l_temp, h_temp, temp, wet, wind, Hecto-pascal Pressure Unit (hpa), visibility, precipitation, and cloud content are the characteristics we used to forecast the Air Quality Index. In our paper, we concentrated on creating models for predicting the Air Quality Index and contrasting Long short term memory (LSTM) and its variants. Bidirectional Long Term Short Term Memory (BiLSTM), Stacked LSTM, and Gated Recurrent Unit (GRU) models. Additionally, as an evolutionary feature selection approach, we included particle swarm optimisation.

Keywords: LSTM, BiLSTM, GRU, PSO

I. INTRODUCTION

"Air pollution" is the presence of dangerous compounds in the Earth's atmosphere that can have a negative impact on climate, the environment, and human health. These chemicals can include particulate matter (PM), volatile organic compounds (VOCs), carbon monoxide (CO), nitrogen dioxide (NO2), sulphur dioxide (SO2), and ozone (O3). Industrial emissions, car exhaust, the burning of fossil fuels, agricultural practises, and natural occurrences like dust storms and wildfires are the main contributors to air pollution. Air pollution is determined by AQI (Air Quality Index). It offers details on the severity of air pollution and any potential negative effects on human health. Predicting the AQI requires predicting the air quality conditions for a specific location at a given time. Urban development and public health depend on accurate AQI prediction. It facilitates long-term environmental planning, enables government agencies to adopt efficient pollution control measures, and assists people in making educated decisions regarding outdoor activities.

Some of the traditional techniques or mathematical models like Regression models, such as linear regression, multiple regression, or polynomial regression, Support Vector Machines (SVM) are

frequently used in traditional methods for AQI prediction. These methods take into account variables like weather information, pollutant concentrations, emission sources, and historical trends. These methods do, however, have some drawbacks. They frequently make the assumption that variables are linearly related, which may not accurately reflect the complicate dynamics of air pollution. Additionally, non-linear and time-dependent patterns in the data are difficult for traditional approaches to handle, making it difficult to effectively forecast abrupt changes or short-term swings in air quality.

Artificial neural networks have advanced quickly in recent years, and because of their strong non-linear fitting and learning capabilities, they have been utilised to create air quality prediction models that have produced more accurate and workable findings. Comparing the correlation coefficient R reveals that the artificial neural network's projected values are fairly similar to the actual values. and the difference between the expected and actual values' root mean square error (RMSE). Therefore, decision-makers can use this network model to forecast local AQI in real time..

Updated versions of Recurrent neural network (RNN) architectures such as Long Short-Term Memory (LSTM) works better for time series sequential data. LSTM networks to handle time-series data particularly well makes them suitable for AQI prediction is mostly due to its ability to find . Unlike conventional techniques, LSTM can automatically identify complex dependencies and patterns in the data, including time-dependent dynamics and non-linear interactions. This makes it possible for LSTM models to include both short-term variations and long-term trends in air pollution, producing predictions that are more precise. Additionally, LSTM can handle variable-length sequences and efficiently use past data to forecast the future. LSTM is a well-liked option for time-series forecasting jobs, including AQI prediction, due to its adaptability and strength.

II. LITERATURE SURVEY

Y. Jiao, Z. Wang and Y. Zhang in [1] proposed that the application of LSTM models in environmental prediction has gained attention due to their ability to capture complex temporal patterns. Evaluations of LSTM-based prediction models have shown promising results in accurately forecasting the Air Quality Index (AQI), suggesting their potential in enhancing air quality monitoring and management practices.

T. Manna and A. Anitha, in [2]. This paper proposes a stacked-LSTM model for predicting air pollution levels in the 15 most polluted cities of India, including Delhi, Uttar Pradesh, Rajasthan, and Haryana. A dataset comprising 69 air quality observing stations from 2015 to 2020 was used to train and test the model, and its forecasting ability was validated against the 2021 air quality data. Comparative analysis with other deep learning techniques demonstrates that the proposed model achieves a remarkable 92 percent prediction accuracy, outperforming alternative approaches.

A. Barve, V. Mohan Singh, S. Shrirao and M. Bedekar, in [3]. The paper investigates the effects of integrating a parallel DNN with the LSTM cell, providing insights into the benefits of this approach compared to using the cell alone.

Y. Zhongjie, W. Shengwei and W. Ze, in [4]. This paper proposes the utilization of the CS algorithm for parameter optimization in predicting AQI. The optimized parameters obtained from the CS algorithm are then used as input parameters for the LSTM model, resulting in the development of the CS-LSTM model for AQI prediction. The experiment conducted on daily AQI data from Rizhao City demonstrates that the CS-LSTM model outperforms the basic model, exhibiting improved prediction accuracy.

L. Zhoul, M. Chenl and Q. Ni, in [5]. The paper introduces the Prophet-SVR hybrid model and the Prophet-LSTM hybrid model to enhance the prediction accuracy of the Prophet model. The findings indicate that the Prophet-LSTM hybrid model performs the best, surpassing the individual models in terms of prediction accuracy and demonstrating significant advantages in air quality index prediction.

H. Wu, Y. Zhang, Q. Zhang, K. Duan, Y.

Lin and S. Du in [6]. This paper proposes an innovative approach that combines an improved gray wolf algorithm (GWO) with a long short-term memory network (LSTM) to predict the Air Quality Index (AQI). To address the limitations of traditional LSTM networks prone to local optima, the connection weights of LSTM are iteratively updated using an enhanced gray wolf algorithm.

T. Madan, S. Sagar and D. Virman in [7]. This paper Machine learning algorithms have been employed to determine the AQI, utilizing datasets from sources like Kaggle and air quality monitoring sites, divided into training set and testing set. Various algorithms, including Linear Regression, Decision Tree, Random Forest, Artificial Neural Network, and Support Vector Machine, have been explored in research to predict air quality. However, achieving accurate results remains a challenge in this field.

S. Wei, H. Xi, K. Zhang, Y. Yun and H. Li in [9], In order to improve the reliability and accuracy of daily Air Quality Index (AQI) forecasting, this paper introduces an optimized LSTM model using the improved Grey Wolf Algorithm (IGWO). The collected data is preprocessed and divided into training and testing sets. To enhance the diversity of the population, the Tent Chaotic Sequence is employed to generate an initial population.

H. Chen, M. Guan and H. Li, in [10]. This paper proposes a method based on dual LSTM (Long Short-Term Memory) model. Firstly, a single prediction model using Seq2Seq (Sequence to Sequence) technology is established to independently predict each component of air quality data, treating them as individual time series. Then, a multi-factor prediction model utilizing LSTM with an attention mechanism is employed, incorporating factors such as neighboring station data and weather information. Finally, the XGBoosting (eXtreme Gradient Boosting) tree is utilized to integrate the two models, getting the final prediction results by gathering the predicted values of optimal subtree nodes.

III. DATASET DESCRIPTION

The Data-set that we considered consists of two cities Shanghai, Wuhan. Features of the data-set that we considered are PM2.5, PM10, SO2, NO2, O3, CO, l_temp, h_temp, temp, wet, wind, Hecto-

pascal Pressure Unit (hpa), visibility, precipitation, and cloud content. The target variable is AQI (Air Quality Index). Size of shanghai dataset considered is (488,17) with 488 data points and 17 features. Size of Wuhan dataset considered is (478,17) with 478 data points and 18 features. Shanghai data set is given below in Figure 1. Wuhan dataset is given in Figure 2.

	date	AQI_24h	PM2.5_24h	PM10_24h	SO2_24h	NO2_24h	O3_24h	CO_24h	l_temp	h_temp	wet	wind	hpa	visibility	precipitation	cloud	
0	2021/1/1	55.53	37.89	58.21	9.74	57.11	22.37	0.83	-2.7	3.9	0.1	45	0.1	1030	22.5	0.0	10
1	2021/1/2	72.88	53.17	64.71	9.29	65.96	26.58	0.90	-0.9	7.0	2.8	53	0.3	1030	14.7	0.0	8
2	2021/1/3	65.39	42.30	51.26	5.22	53.52	32.09	0.62	2.0	10.2	6.6	74	0.3	1030	17.1	0.0	10
3	2021/1/4	42.12	20.71	41.80	4.79	51.71	34.17	0.45	7.2	12.9	9.6	75	0.3	1027	17.7	0.0	36
4	2021/1/5	48.92	32.12	45.21	5.71	31.25	53.38	0.54	5.0	9.1	7.5	69	0.6	1029	14.9	0.0	76
...
473	2022/4/19	55.62	38.29	55.62	8.25	25.58	122.42	0.81	13.1	21.8	17.4	53	0.4	1018	16.8	0.0	46
474	2022/4/20	71.29	43.71	70.50	10.25	23.42	148.54	0.87	14.1	24.4	19.2	53	0.7	1015	13.7	0.0	48
475	2022/4/21	66.50	45.88	57.25	7.58	17.25	145.33	0.91	14.9	26.2	19.6	74	0.6	1011	9.4	3.3	35
476	2022/4/22	53.08	36.54	47.67	10.12	24.42	116.88	0.94	17.1	28.9	22.6	67	0.5	1010	13.5	0.0	61
477	2022/4/23	33.21	16.96	17.25	5.54	14.96	90.46	0.79	17.6	21.9	19.1	97	0.6	1010	11.9	29.6	89

478 rows x 17 columns

Fig. 1: Shanghai Data set

	date	AQI	PM2.5	PM10	SO2	NO2	O3	CO	l_temp	h_temp	wet	wind	hpa	visibility	precipitation	cloud	
0	2019/9/1	57	32	49	8	45	95	1.0	25.7	20.2	22.8	87.0	0.1	1011.0	11.4	1.0	100.0
1	2019/9/2	100	35	54	11	38	159	1.1	30.6	20.7	25.5	74.0	0.6	1012.0	12.3	0.1	88.0
2	2019/9/3	74	23	39	9	25	128	0.9	30.6	22.4	26.3	64.0	1.0	1010.0	24.6	0.0	70.0
3	2019/9/4	91	25	44	9	27	149	0.9	32.2	21.1	26.9	63.0	1.6	1007.0	17.6	0.0	30.0
4	2019/9/5	124	29	54	12	38	186	1.2	32.0	23.4	27.1	67.0	1.0	1005.0	21.6	0.0	64.0
...
483	2020/12/27	216	166	202	21	100	71	1.6	14.8	2.1	7.8	78.0	0.3	1016.0	1.8	0.0	24.0
484	2020/12/28	186	140	168	15	78	82	1.6	16.8	1.7	9.3	81.0	0.8	1014.0	1.7	2.7	27.0
485	2020/12/29	79	58	65	5	29	41	1.1	9.4	0.3	3.2	87.0	6.2	1026.0	3.9	11.7	90.0
486	2020/12/30	43	26	43	7	24	62	0.5	3.2	-3.5	-0.5	58.0	2.7	1037.0	19.0	0.0	14.0
487	2020/12/31	50	19	36	8	40	51	0.5	4.3	-7.6	-2.0	55.0	0.3	1034.0	23.1	0.0	16.0

488 rows x 17 columns

Fig. 2: Wuhan Data set

Missing Values:

- Shanghai dataset doesn't contain any NULL/NAN values in the dataset.
- Wuhan dataset have NULL/NAN values as given l_temp has 2,h_temp has 2,temp has 2,wet has 2,wind has 2,hpa has 24,visibility has 24, precipitation has 2, cloud has 69 NULL values.

Data description: The statistical observations of each feature in wuhan dataset is given in figure 3 The statistical observations of each feature in wuhan dataset is given in figure 18

IV. EXPLORATORY DATA ANALYSIS

Exploratory Data Analysis (EDA) is a fundamental step in data science that involves understanding the dataset statistically. EDA is essential for feature selection and engineering since it evaluates the variables' relevance to the target and their

	AQI_24h	PM2.5_24h	PM10_24h	SO2_24h	NO2_24h	O3_24h	CO_24h	l_temp
count	478.000000	478.000000	478.000000	478.000000	478.000000	478.000000	478.000000	478.000000
mean	48.276464	28.621611	44.669979	5.683975	34.173013	65.713452	0.638117	13.903766
std	23.510271	17.200097	27.131350	1.720962	16.355499	24.880974	0.177690	8.484342
min	10.170000	3.580000	8.460000	3.960000	4.090000	8.500000	0.350000	-7.000000
25%	32.042500	15.520000	27.760000	4.627500	22.400000	48.592500	0.510000	7.200000
50%	43.620000	24.480000	37.750000	5.170000	30.980000	64.310000	0.605000	12.500000
75%	60.562500	36.987500	54.770000	6.110000	42.110000	82.115000	0.720000	22.500000
max	245.290000	112.790000	309.880000	20.960000	113.670000	148.540000	1.460000	29.200000

Fig. 3: Shanghai Data set statistical interpretation

	AQI	PM2.5	PM10	SO2	NO2	O3	CO	l_temp
count	488.000000	488.000000	488.000000	488.000000	488.000000	488.000000	488.000000	488.000000
mean	75.891393	39.415984	62.221311	8.631148	39.82582	92.526639	0.884016	21.279630
std	31.399174	23.819330	32.722594	3.418036	20.93307	46.743383	0.220952	9.007133
min	20.000000	5.000000	9.000000	4.000000	10.00000	5.000000	0.400000	2.400000
25%	51.000000	24.000000	38.000000	6.000000	24.00000	55.000000	0.700000	13.800000
50%	73.000000	34.000000	57.000000	8.000000	34.00000	88.000000	0.900000	22.200000
75%	95.250000	49.000000	81.000000	10.000000	49.25000	124.250000	1.000000	28.625000
max	223.000000	173.000000	216.000000	24.000000	124.00000	222.000000	1.800000	37.300000

Fig. 4: Wuhan Data set statistical interpretation

effects on it.

EDA also helps in identifying outliers that may have an impact on model performance. EDA offers insightful information and directs future analysis and decision-making by revealing patterns and linkages. EDA verifies the adequacy of the assumptions used in statistical models. EDA-derived visualisations and summaries help stakeholders understand complex information more effectively. Overall, EDA is an integral part of the data science workflow, enhancing the accuracy and reliability of subsequent data processing, modeling, and evaluation.

Visualisation plots:

The Figure 5 given below is useful for identifying the 'AQI_24h' variable's trend or trends over time. The dates are represented on the x-axis, and the air quality index (AQI) value for a 24-hour period is shown on the y-axis. The line connecting the data points shows how the AQI has changed over time. It makes it simpler to comprehend the dates by rotating the x-axis labels.

This graph can gives better idea about the variations in the AQI readings over time and assist in spotting any time dependent patterns or trends in the air quality data. It is especially helpful for time series analysis and examining how changes

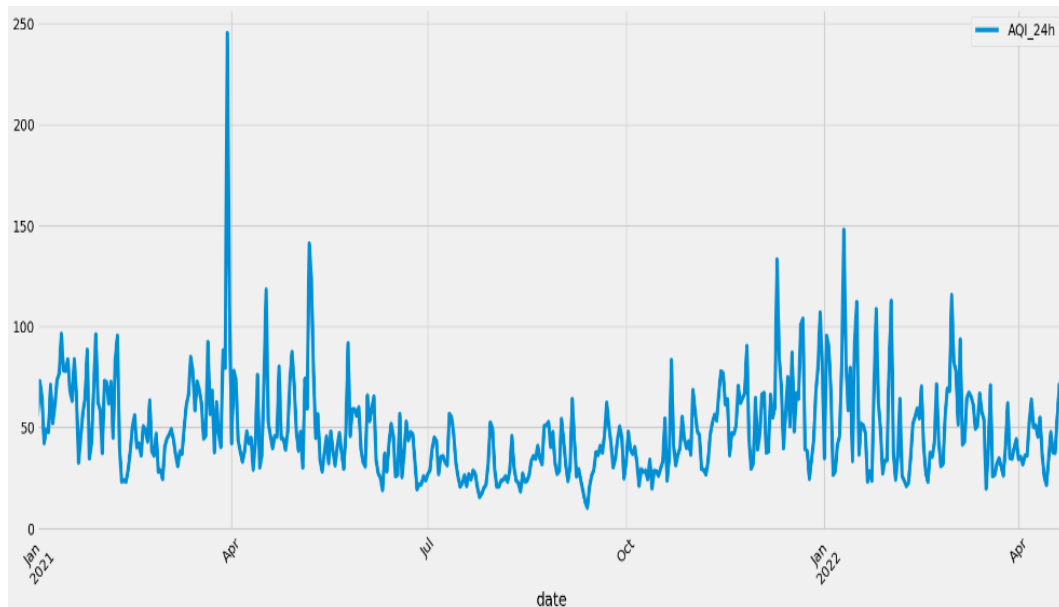


Fig. 5: Wuhan Data set statistical interpretation

in air quality have occurred over the course of the observational period.

1.Box Plots:

Box plots are crucial tool in statistical analysis and data visualization, giving valuable insights into the distribution, central tendency, and variability of a dataset. The value of box plots resides in their capacity to provide a clear and insightful summary of complex data. A box plot has a number of essential elements. The box firstly shows the interquartile range (IQR), which is consists of the middle 50% of the data. The median, shown by the line inside the box, denotes the dataset's centre tendency. The data range, excluding outliers, is displayed by the whiskers that extend from the box. If there are any outliers, they are shown as separate points outside the whiskers. Box plots offer a clear visualisation of the dispersion, skewness, and likely existence of outliers in a dataset by presenting these attributes. They make it simple to compare several datasets or groups within a dataset, which makes it easier to spot patterns, trends, and variations in data distributions. Box plots are a crucial part of efficiently analysing statistical data since they are a powerful tool for exploratory data analysis. The boxplots for monthly average AQI is given in figure- 6 and the boxplots for yearly average AQI is given in figure- 7

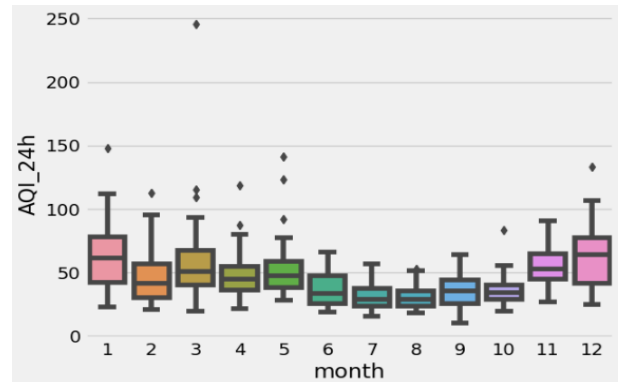


Fig. 6: Box plot Monthly AQI Shanghai data

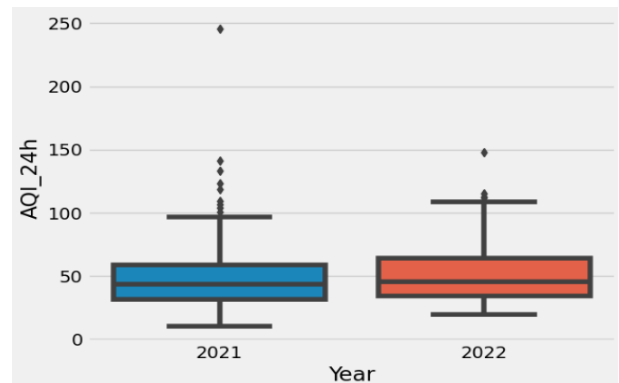


Fig. 7: Box Plot Yearly AQI Shanghai data

2.Correlation analysis:

Correlation analysis surrounds various techniques to explore and measure the relationship between variables. Some different ways of con-

ducting correlation analysis include:

- 1) **Pearson Correlation:** This method assesses the linear relationship between two continuous variables. The Pearson correlation coefficient, which has a range of -1 to +1, is calculated. Perfect positive correlation is represented by a value of 1, perfect negative correlation by a value of -1, and zero indicates no linear correlation.
- 2) **Spearman correlation:** This method measures the monotonic relationship between variables, in contrast to the Pearson correlation. It is employed when the relationship is non-linear or the variables are not regularly distributed. It rates the data and uses the ranks to calculate the correlation.
- 3) **Kendall's Tau:** Kendall's Tau coefficient assesses the degree and direction of the monotonic relationship between variables, much like the Spearman correlation does. If you're working with ranked or ordinal data, it's especially helpful.

The correlations and dependencies between variables are frequently represented visually in correlation analysis using heatmaps. They offer a useful tool for understanding the magnitude and direction of relationships between various variables. The corresponding heat map for Shanghai dataset is given in Figure 8. The



Fig. 8: Heat Map Shanghai data

Heatmap for Wuhan dataset is given below Figure 9.



Fig. 9: Heat Map Shanghai data

3.Examining Seasonality patterns:

Seasonality in time series data refers to a regular and predictable pattern of variation that occurs at specific time intervals. It is characterized by recurring fluctuations or trends that repeat within a given time frame, such as days, weeks, months, or years. These patterns can be influenced by various factors, such as weather conditions, holidays, or economic cycles. By identifying and understanding seasonality in time series data, analysts can gain insights into the cyclic behavior and make more accurate predictions or forecasts. The seasonality identification is due the weather and climatic conditions and the intensity of seasonality is low.

The seasonality auto correlation graph of our time series data is given in Figure 10.

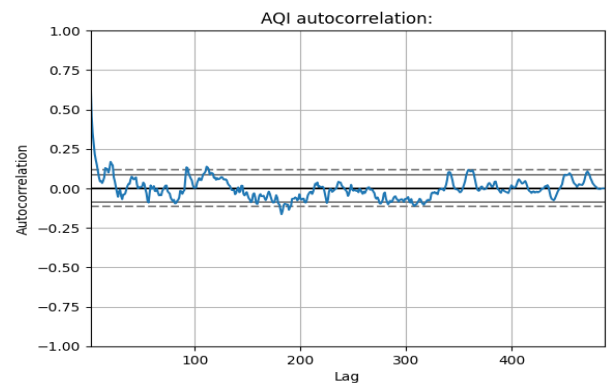


Fig. 10: Seasonality Auto Correlation Plot

V. METHODOLOGY

After going through Exploratory Data Analysis, Feature engineering is to be done to improve respective model. Feature selection plays a major role in feature engineering which takes out best features from the given dataset features. Using

Feature selection methods, the notable advantages are Improved model performance, Enhanced interpretability, Faster training and inference, Noise reduction, Data collection and storage efficiency. Feature selection can also be done using Evolutionary Algorithms like Particle swarm optimisation.

Particle Swarm Optimisation:

Particle Swarm optimisation algorithm is a global heuristic method developed based on Swarm intelligence. It is an evolutionary algorithm that is extracted from nature's behaviour. The major change in particle swarm optimisation when compared to basic swarm optimisation is the co-operation of each member or individual which we call a particle. The simple and optimised approach obtained is due to individual contribution. Due to this Particle Swarm Optimisation Algorithm is being widely used in fields of optimisations of certain complex functions, signal processing, neural network training.

Major Principles of particle swarm optimisation:

- 1) Maintaining inertia.
- 2) Updating the positions according to the most optimal position of particle in each iteration.
- 3) Updating the positions according to the most optimal position of swarm in each iteration.

All particles in the swarm maintains position, speed and updates it through each iteration. Process of Particle swarm optimisation is given in fig

The speed V of particle i after k iterations is a_{id}^k and position X of particle i is a_{id}^k . Then the velocity update after k iterations will be:

$$v_{id}^{k+1} = v_{id}^k + c_1 r_1^k (pbest_{id}^k - x_{id}^k) + c_2 r_2^k (gbest_d^k - x_{id}^k)$$

here, $pbest_{id}^k$ refers to optimistic position after k iterations. $gbest_{id}^k$ refers to dimension quantity of swarm at optimal position.

$$x_{id}^{k+1} = k_{id}^x + v_{id}^{k+1}$$

The maximum value of v_{dmax} indicates that the optimal solution is very far away from the present position. The minimum value of v_{dmax} indicates that it is the local optimal solution. The speed of the particle lies in the space of $-v_{dmax}$ to v_{dmax} .

Pseudo Code of Particle Swarm Optimisation:

Algorithm 1 Particle Swarm Optimization

```

1: Initialize  $X_i, V_i, iteration, pbest, gbest$ 
2: Generate random particles (P) from features set
3: for each Particle  $i$  do
4:   Calculate fitness function  $f_i$  for each feature
5:   Update  $pbest, gbest$  values
6: end for
7: while  $iteration$  do
8:   for each particle in features do
9:     Update  $X_i, V_i$ 
10:    if  $X_i > threshold$  then
11:       $X_i = threshold$ 
12:    end if
13:    Calculate fitness function  $f_i$  of feature  $i$ 
14:    Update  $pbest, gbest$  values
15:  end for
16: end while

```

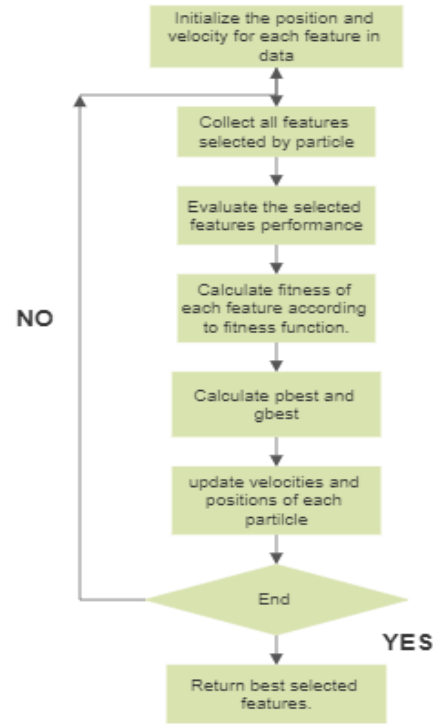


Fig. 11: Process of Particle Swarm Optimisation

Figure- 14 shows the process of Particle swarm optimisation.

Advantages of Particle Swarm Optimisation:

- 1) Simple math and simple calculations.
- 2) It doesn't have mutation and overlapping conditions as only the optimistic particles can transmit information to further generations.

Disadvantages of Particle Swarm Optimisation:

- 1) PSO algorithm work only under the consideration of co-ordinate system.

Feature Selection using Particle Swarm Optimisation

The process of finding the best subset of features that maximises a fitness function is known as feature selection via particle swarm optimisation. A high-level explanation of how PSO can be applied to feature selection is given below:

- 1) Start-up: Create a swarm of particles, each of which stands for a potential feature subset.
- 2) Fitness Assessment: Using a fitness function that gauges the calibre of the appropriate feature subset, assess each particle's fitness.
- 3) Particle Movement: Each particle's velocity and position are updated based on its own knowledge and the swarm's determination of the world's best place.
- 4) Iteration: Repeat the fitness assessment and particle movement phases until a termination criterion is satisfied, for a predetermined number of iterations.
- 5) Establish a stopping condition, such as reaching a set number of iterations or achieving convergence, as the termination criterion.
- 6) Track and update the global best position, which is the best feature subset that any particle in the swarm has discovered.
- 7) The feature subset that corresponds to the particle with the highest fitness value should be chosen as the final solution after all iterations have been completed.
- 8) Create a fitness function that accurately reflects the trade-off between the complexity (such as the amount of features chosen) and the quality (such as classification or regression performance) of the feature subsets.
- 9) Feature Subset Encoding: Represent the position of each particle as a binary string, where each bit denotes whether a certain feature is present or not.
- 10) Update the particle's speed according to its present speed, its own personal best position, and the world's best position discovered by the swarm.
- 11) Define the neighbourhood structure within the swarm that controls how the movement

of the particles is influenced by one another.

- 12) Performance Evaluation: Depending on the kind of problem, evaluate the performance of the chosen feature subset using the appropriate evaluation metrics, such as accuracy, precision, recall, or mean squared error.
- 13) Verification: Verify the performance of the chosen feature subset using hypothetical data to check for generalizability.
- 14) Comparison and Validation: To evaluate the success of PSO-based feature selection, compare its outcomes to those of other feature selection techniques or baseline models.

Basic LSTM Model

In 1997, Hochreiter and Schmidhuber proposed the LSTM in order to solve the issues of "gradient disappearance" and "gradient explosion" in RNN models. The basic units or fundamental blocks of the LSTM model are one or group of memory modules. Each module contains memory units and gates to control over the system's information flow.

LSTM is an updated version of RNN (Recurrent Neural Networks). As it has internal cell units, it is able to memorise all time series and store, move outputs to further cells and evaluate it at each cell. Despite an RNN, LSTM has the capability to memorise time series because each cell unit comprises three different logic gates which are based on a sigmoid neural network layer, they are input gate, an output gate, and a forgetting gate, which helps in evaluation, transferring output to further cells. The basic flow of the LSTM model is given in Figure 12.

The mathematical model on which Long short term memory model works is given below:

$$i_t = \sigma(W_i[h_{t-1}, x_t]) + b_i$$

$$f_t = \sigma(W_f[h_{t-1}, x_t]) + b_f$$

$$o_t = \sigma(W_o[h_{t-1}, x_t]) + b_o$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_c[h_{t-1}, x_t] + b_c)$$

$$h_t = o_t \tanh(c_t)$$

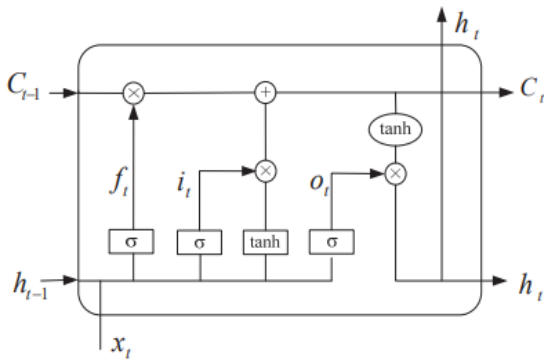


Fig. 12: Flow of LSTM model

where i , f , o , c are input gates, forgetting gates, output gates, cell states. while W , B are corresponding weights of coefficient matrices, terms of off-sets. σ is the sigmoid activation function and \tanh is hyperbolic tan activation function.

If input is represented as 0, then no data is doesn't pass through it and if it is displayed as 1, all data is allowed to go through it. The number of data segments that can flow through each unit is specified by a value that each layer of a sigmoid network can generate between 0 and 1.

Forgetting gate generates data between 0 and 1, where 0 denotes total disregard and 1 denotes total reserve.

Storage gates helps to store the outputs obtained from each cell and can easily retrieve form it when required. They consist of a \tanh layer and a sigmoid layer (the input gate). After the sigmoid layer chooses the values that need to be altered, the \tanh layer starts maintaining vectors of new candidate values and adds them to the cell unit state.

Step 1: Applying the forward calculation method to calculate the output value of LSTM cells;

Step 2: Reversing the calculation of each LSTM cell's error term, which includes two routes for reverse propagation based on the time and network level;

Step 3: Calculating each weight's gradient in accordance with the associated error term;

Step 4: Update the weights using gradient-based optimisation methods (like Adam).

BiLSTM model

A BiLSTM (Bidirectional Long Short-Term Memory) model is a type of recurrent neural network (RNN) architecture that combines the power of both forward and backward information flow. It is frequently used for applications involving sequence modelling, such as sentiment analysis, named entity recognition, machine translation, and speech recognition in natural language processing (NLP).

The LSTM cell serves as the fundamental building unit of a BiLSTM model. An LSTM cell employs input, forget, and output gates in conjunction to capture long-term dependencies in sequential data. These gates regulate the information flow within the cell, enabling it to retain or lose particular information over time.

A conventional LSTM simply considers the past data while determining the hidden state at each time step. A BiLSTM model, on the other hand, employs two independent LSTMs, one of which processes the input sequence forward from beginning to end and the other of which processes the sequence backward from end to beginning. The final result is created by concatenating or combining the outputs from these two LSTMs.

A BiLSTM model can represent dependencies from both past and future contexts by incorporating both forward and backward information. This enables the model to comprehend the input sequence more thoroughly, which improves its suitability for jobs requiring the collection of contextual information from both directions.

The typical architecture of a BiLSTM model involves the following steps:

1. Input Encoding: The input sequence is represented as a sequence of word embeddings or numerical vectors.
2. Forward LSTM Pass: The forward LSTM processes the input sequence from the beginning to the end, updating its hidden state at each time step.
3. Backward LSTM Pass: The backward LSTM processes the input sequence from the end to the

beginning, updating its hidden state at each time step.

4. Output Combination: The outputs from the forward and backward LSTMs are combined, often by concatenation or some other merging operation, to form the final output sequence.

5. Prediction Layer: The final output sequence is passed through one or more additional layers (e.g., fully connected layers) to make predictions or generate the desired output.

During the training process, the parameters of the BiLSTM model are learned using gradient-based optimization algorithms like back-propagation through time (BPTT) or variants of stochastic gradient descent (SGD). The model learns, often through supervised learning with labelled data, how to change its parameters to minimise a certain loss function.

Overall, the BiLSTM model performs better on a variety of sequence modelling tasks thanks to its capacity to take into account both past and future context.

GRU model

A GRU (Gated Recurrent Unit) model is a type of recurrent neural network (RNN) architecture used for sequence modeling tasks. Although it has a simpler design, it is related to LSTM (Long Short-Term Memory). An update gate, a reset gate, a candidate activation, and a hidden state make up a GRU cell. How much of the prior hidden state should be kept and how much of the fresh input should be integrated are determined by the update gate. How much of the prior hidden state should be forgotten is decided by the reset gate. Based on the current input and the previous hidden state and taking the reset gate into account, the candidate activation determines a new hidden state.

The key components of a GRU cell are as follows:

1. Update Gate (z): It determines how much of the previous hidden state should be preserved and how much the new input should be integrated into the current state. It controls the flow of information from the previous time step to the current time step.

2. Reset Gate (r): It decides how much of the previous hidden state should be forgotten or ignored. It allows the model to capture short-term dependencies in the input sequence.

3. Candidate Activation (h^-): It calculates a new candidate hidden state based on the current input and the previous hidden state, taking into account the reset gate.

4. Current Hidden State (h): It combines the candidate activation with the update gate, producing the current hidden state. It represents the memory of the GRU cell at the current time step

During training, the GRU model learns to adjust its parameters through back-propagation to minimize a specified loss function. It is suitable for jobs like sentiment analysis and machine translation since it recognises both temporary and permanent dependencies in the input sequence. Due to its more straightforward architecture than LSTM, the GRU model is more efficient computationally and doesn't complicate while training.

VI. EXPERIMENTS AND RESULTS

The Procedure of AQI prediction model follows these steps:

- 1) Exploratory Data Analysis(EDA)
- 2) Feature Selection
- 3) Feature Scaling
- 4) Training the LSTM, BiLSTM, GRU models
- 5) Fitting the models with proper optimiser
- 6) Performance evaluation.

Feature Selection Using Particle Swarm Optimisation The whole process of model designed is given in figure 13 Particle Swarm Optimisation is applied to the data by adding functions such as `init_position`, `init_velocity`, `fitness_score` function and so on. The fitness function value with respect to number iteration graph according to the considered data is given in Figure 14.

The output of Feature selection using PSO algorithm is an array of indices of selected features. From our data only two features are mentioned as useless features they are `PM10_24h`, `h_temp`, `visibility`.

Feature Scaling using Min-Max Scaler

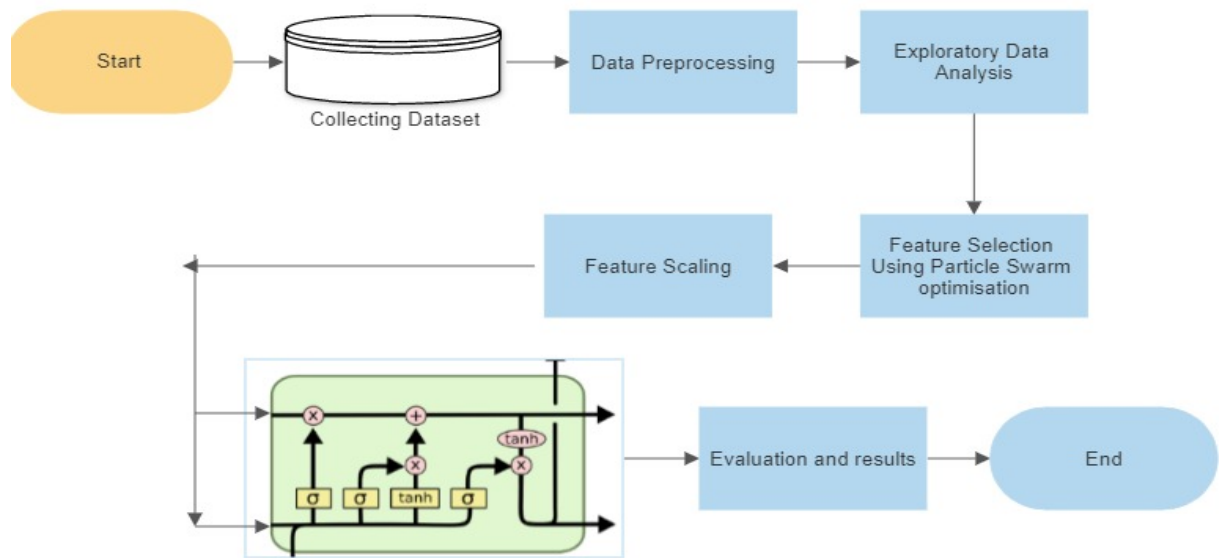


Fig. 13: Flow of implemented process

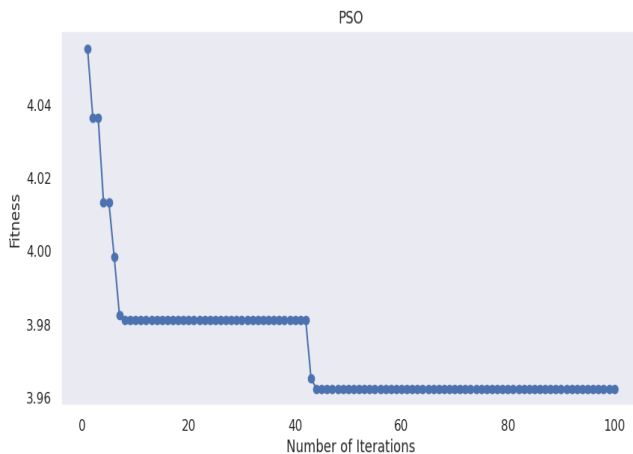


Fig. 14: PSO Graph analysis Fitness vs Number of iterations

Min-Max scaling, commonly referred to as normalisation, is a well-liked data scaling method used to change variables to a particular range, typically between 0 and 1. Each data point must first be subtracted from the minimum value of the variable before being divided by the range, which is the range's range between maximum and minimum values.

This scaling strategy is especially helpful when maintaining the relative proportions and relationships between variables is crucial. It facilitates fair comparisons between variables and prevents the dominance of extreme values by scaling the data to a defined range. By establishing a consis-

tent scale for the variables, improving the performance of various algorithms, and bring up a better knowledge of the data, min-max scaling enables better analysis, modelling, and interpretation.

Training LSTM model

The model you gave is a sequence-to-sequence LSTM model, which is frequently used for language translation, time series forecasting, and sequence prediction. It comprises of a RepeatVector layer, a TimeDistributed layer with a Dense layer, and two LSTM layers of 200 units each.

Here's an explanation of each component of the model:

- 1) The input sequence is processed by the first LSTM layer, which has 200 units and relu activation. It has 200 units, which stand in for the LSTM's memory cells. The 'relu' activation function is used to add non-linearity to the LSTM calculations. The input structure is given as (n_steps_in, n_features), where n_steps_in denotes the sequence's total number of time steps and n_features denotes the number of features present at each step.
- 2) The Repeat Vector layer iteratively repeats the LSTM layer's output n_steps_out times. It keeps the temporal relationship while allowing the output of the first LSTM layer to serve as the input for the second LSTM layer.

- 3) The input sequence is also processed by the second LSTM layer (200 units, relu activation, return_sequences=True). It has 200 units and instead of only returning the final output state, it also returns the output sequence. Setting return_sequences=True indicates this. The output sequence's additional temporal information is crucial for capturing the dynamics of the input sequence.
- 4) In the Time Distributed layer, the Dense layer is applied individually to each time step of the output sequence from the previous LSTM layer. The number of features in the output sequence is represented by the n_features units of the Dense layer. Non-linearity is introduced, and the final mapping to the desired output format is carried out.
- 5) Overall, the LSTM layers are able to process the input sequence and capture the temporal dependencies thanks to this model architecture. The initial LSTM layer's output sequence is repeated n_steps_out times thanks to the Repeat Vector layer. This repeated sequence is subsequently processed by the second LSTM layer, which also records the temporal dynamics. The final output sequence is created by applying a non-linear transformation separately to each time step in the Time Distributed layer with the Dense layer.

Training BiLSTM model

- 1) By using the Bidirectional layer, the BiLSTM model processes the input sequence in both the forward and backward directions. This improves the model's comprehension of the temporal connections within the sequence by enabling it to capture information from both past and future contexts. While the backward LSTM layer records patterns from the future to the past, the forward LSTM layer does the opposite.
- 2) The model's overall architecture doesn't change. To achieve the necessary output sequence length, the BiLSTM layer's output is repeated using the RepeatVector layer. To keep the information flowing in both directions, the second LSTM layer is likewise replaced with a bidirectional layer. The Dense

layer is then applied to each time step of the output sequence by the TimeDistributed layer.

Overall, using a Bidirectional LSTM model can improve the model's ability to capture complex temporal dependencies and can be particularly useful in tasks where past and future context are equally important.

Training GRU model

- 1) The GRU is a variation of the LSTM architecture that simplifies the cell state by combining the forget and input gates into a single update gate. As a result, the design is more streamlined and has fewer parameters than the LSTM due to the decreased number of gates.
- 2) In order to keep the same number of units and activation function, the LSTM layer must be replaced with a GRU layer.
- 3) The model's overall architecture doesn't change. The output of the GRU layer is repeated using the RepeatVector layer to get the desired output sequence length. In order to guarantee that the output sequence is returned, the second GRU layer is added with the return_sequences=True option. The Dense layer is then applied to each time step of the output sequence by the TimeDistributed layer.
- 4) The model, which has a more straightforward architecture than LSTM, can nonetheless detect temporal dependencies in the input sequence by using GRU layers.

VII. MODEL RESULTS

Results of LSTM on Shanghai Dataset

Step	t+1	t+2	t+3
m=6	21.61	23.15	24.20
m=8	22.63	22.92	23.37
m=12	20.64	21.04	22.99

We can conclude from results that the choice of the number of LSTM units (m) can impact the predicted AQI values, with the model having 12 LSTM units generally provides lower predictions compared to the models with 6 or 8 LSTM units. The time step also influences the

predicted AQI values, with the third time step generally yielding higher predictions compared to the first and second time steps. These results demonstrate the ability of LSTM models in order to capture all temporal patterns making predictions from those on air quality based on historical data

Results of GRU on shanghai Dataset

Comparative results of GRU on Shanghai			
Step	t+1	t+2	t+3
m=6	18.87	20.14	22.61
m=8	19.87	21.92	21.97
m=12	20.56	21.66	22.21

when the GRU model is used the results show the predicted Air Quality Index (AQI) values using the GRU model on the Shanghai dataset for different time steps and number of GRU units. The specific time step and the number of GRU units (m) impact the predicted AQI values. The GRU model with 12 GRU units tends to produce higher AQI predictions compared to the models with 6 or 8 GRU units. Additionally, the third time step generally yields higher predictions compared to the first and second time steps. These results highlight the GRU model's ability to capture temporal patterns and provide predictions on air quality based on historical data

Results of BiLSTM on shanghai

Comparative results of BiLSTM on Shanghai			
Step	t+1	t+2	t+3
m=6	19.61	22.56	22.61
m=8	22.42	22.92	22.97
m=12	20.56	21.66	22.21

The results illustrate the predicted Air Quality Index (AQI) values using the BiLSTM model on the Shanghai dataset for different time steps and number of BiLSTM units. Similar to the previous models, the specific time step and the number of BiLSTM units (m) influence the predicted AQI values. The BiLSTM model with 8 units consistently produces higher predictions compared to the models with 6 or 12 units. The second time step tends to yield higher predictions compared to the first time step, while the third time step shows mixed results. These findings highlight the effectiveness of the BiLSTM model in capturing temporal patterns and making

predictions on air quality based on historical data.

Results of LSTM on Wuhan

Comparative results of LSTM on Wuhan Dataset			
Step	t+1	t+2	t+3
m=6	20.35	23.15	24.20
m=8	22.63	22.92	23.37
m=12	20.64	21.04	22.99

This table indicates the predicted Air Quality Index (AQI) values using the LSTM model on the Wuhan dataset for different time steps and number of LSTM units. The choice of the number of LSTM units (m) influences the predicted AQI values, with the model having 8 LSTM units generally producing higher predictions compared to the models with 6 or 12 LSTM units. Moreover, the third time step tends to yield the highest predictions, while the first and second time steps show varying results. These findings highlight the LSTM model's ability to capture temporal patterns and provide predictions on air quality based on historical data for the Wuhan dataset.

Results of BiLSTM on Wuhan Dataset

Comparative results of BiLSTM on Wuhan			
Step	t+1	t+2	t+3
m=6	18.35	21.15	22.20
m=8	20.63	21.82	22.35
m=12	20.64	21.04	22.99

Results illustrate the predicted Air Quality Index (AQI) values using the BiLSTM model on the Wuhan dataset for different time steps and number of BiLSTM units. The choice of the number of BiLSTM units (m) influences the predicted AQI values, with the model having 8 BiLSTM units generally producing higher predictions compared to the models with 6 or 12 BiLSTM units. Furthermore, the third time step tends to yield the highest predictions, while the first and second time steps show varying results. These findings highlight the effectiveness of the BiLSTM model in capturing temporal patterns and providing predictions on air quality based on historical data for the Wuhan dataset.

Results of GRU on Wuhan Dataset

Comparative results of GRU on Wuhan Dataset			
Step	t+1	t+2	t+3
m=6	17.35	19.15	21.20
m=8	20.63	21.92	23.37
m=12	20.84	21.04	23.99

Results shows the predicted Air Quality Index (AQI) values using the GRU model on the Wuhan dataset for different time steps and number of GRU units. The choice of the number of GRU units (m) influences the predicted AQI values, with the model having 8 GRU units generally producing higher predictions compared to the models with 6 or 12 GRU units. Additionally, the third time step tends to yield the highest predictions, while the first and second time steps show varying results. These findings highlight the effectiveness of the GRU model in capturing temporal patterns and providing predictions on air quality based on historical data for the Wuhan dataset.

The respective loss value graph in case of Shanghai dataset is given in Figure 15

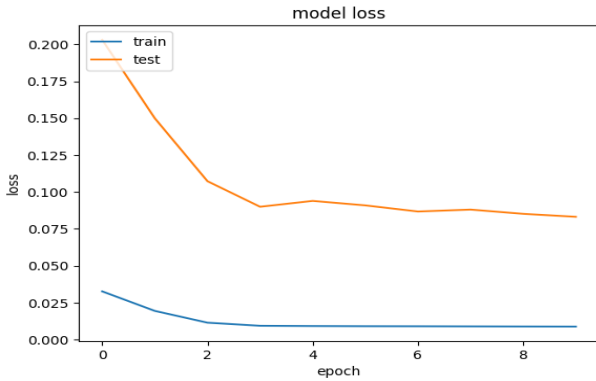


Fig. 15: Losses per echos Graph for shanghai dataset

The variation comparison graph of AQI shanghai data set is given in Figure 16 Similarly, The

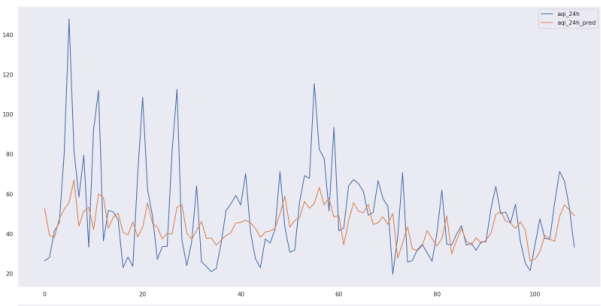


Fig. 16: Comparing Actual AQI vs Predicted AQI for LSTM Shanghai Dataset

variation comparison graph of AQI shanghai data set is given in Figure 17

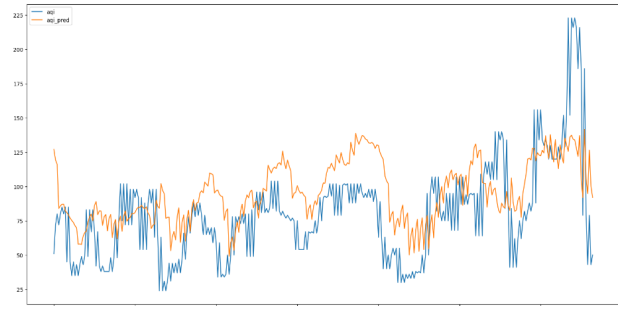


Fig. 17: Comparing Actual AQI vs Predicted AQI for LSTM Wuhan Dataset

All the results recorded are given in table-1.

VIII. CONCLUSION AND FUTURE SCOPE

In conclusion, the AQI (Air Quality Index) prediction models using LSTM, BiLSTM, and GRU have shown promising results in capturing the temporal dependencies and predicting air quality levels. These models have proven to be useful in comprehending the intricate dynamics and patterns found in data on air pollution. These models can offer decision-makers and those who are worried about air pollution useful insights and predictions by taking into account historical air quality measurements.

• Integration of External Factors:

The precision of AQI prediction models can be improved by incorporating outside variables like weather, traffic patterns, and topographical features. Predictions that are more thorough and reliable can be obtained by examining the effects of these elements and including them in the models.

• Integration of External Factors:

The precision of AQI prediction models can be improved by incorporating outside variables like weather, traffic patterns, and topographical features. Predictions that are more thorough and reliable can be obtained by examining the effects of these elements and including them in the models.

• Ensemble Models:

To improve the precision and dependability of AQI forecasts by looking into the usage of ensemble models, which mix many prediction models. Performance can be enhanced

TABLE I: Comparative results of Different Models on Shanghai and Wuhan Datasets

Comparative Results of LSTM, GRU, and BiLSTM					
Dataset	Model	Step	t+1	t+2	t+3
Shanghai	LSTM	m=6	21.61	23.15	24.20
		m=8	22.63	22.92	23.37
		m=12	20.64	21.04	22.99
	BiLSTM	m=6	19.61	22.56	22.61
		m=8	22.42	22.92	22.97
		m=12	20.56	21.66	22.21
	GRU	m=6	18.87	20.14	22.61
		m=8	19.87	21.92	21.97
		m=12	20.56	21.66	22.21
Wuhan	LSTM	m=6	20.35	23.15	24.20
		m=8	22.63	22.92	23.37
		m=12	20.64	21.04	22.99
	BiLSTM	m=6	18.35	21.15	22.20
		m=8	20.63	21.82	22.35
		m=12	20.64	21.04	22.99
	GRU	m=6	17.35	19.15	21.20
		m=8	20.63	21.92	23.37
		m=12	20.84	21.04	23.99

by combining the advantages of many models, including LSTM, BiLSTM, and GRU.

- Fine-Grained AQI Prediction:

While AQI offers a broad overview of air quality, more precise forecasts at localised levels are becoming increasingly necessary. In order to identify pollution major places and guide targeted activities, i

In conclusion, the application of LSTM, BiLSTM, and GRU models for AQI prediction has demonstrated promise in the comprehension and forecasting of air pollution levels. Incorporating external influences, investigating ensemble models, providing real-time updates, fine-grained prediction, enhancing interpretability, and assuring seamless deployment and integration should be the main areas of future research. These developments may lead to AQI predictions that are more precise and useful, which will improve air quality management and public health consequences.

REFERENCES

- [1] Y. Jiao, Z. Wang and Y. Zhang, "Prediction of Air Quality Index Based on LSTM," 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 2019, pp. 17-20, doi: 10.1109/ITAIC.2019.8785602.
- [2] T. Manna and A. Anitha, "Forecasting Air Quality Index based on Stacked LSTM," 2022 IEEE 7th International Conference on Recent Advances and Innovations in Engineering (ICRAIE), MANGALORE, India, 2022, pp. 326-330, doi: 10.1109/ICRAIE56454.2022.10054260
- [3] A. Barve, V. Mohan Singh, S. Shrirao and M. Bedekar, "Air Quality Index forecasting using parallel Dense Neural Network and LSTM cell," 2020 International Conference for Emerging Technology (INCET), Belgaum, India, 2020, pp. 1-4, doi: 10.1109/INCET49848.2020.9154069.
- [4] Y. Zhongjie, W. Shengwei and W. Ze, "Air quality prediction method based on the CS-LSTM," 2022 5th International Conference on Data Science and Information Technology (DSIT), Shanghai, China, 2022, pp. 1-5, doi: 10.1109/DSIT55514.2022.9943922.
- [5] L. Zhoul, M. Chenl and Q. Ni, "A hybrid Prophet-LSTM Model for Prediction of Air Quality Index," 2020 IEEE Symposium Series on Computational Intelligence (SSCI), Canberra, ACT, Australia, 2020, pp. 595-601, doi:

10.1109/SSCI47803.2020.9308543.

[6] H. Wu, Y. Zhang, Q. Zhang, K. Duan, Y. Lin and S. Du, "Prediction of Air Quality Index (AQI) Based on Optimized GWO-LSTM Neural Network," 2022 IEEE 8th International Conference on Computer and Communications (ICCC), Chengdu, China, 2022, pp. 1445-1449, doi: 10.1109/ICCC56324.2022.10065729.

[7] T. Madan, S. Sagar and D. Virmani, "Air Quality Prediction using Machine Learning Algorithms –A Review," 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, India, 2020, pp. 140-145, doi: 10.1109/ICACCCN51052.2020.9362912.

[8] Wu, Z., Zhao, W. Lv, Y. An ensemble LSTM-based AQI forecasting model with decomposition-reconstruction technique via CEEMDAN and fuzzy entropy. *Air Qual Atmos Health* 15, 2299–2311 (2022). <https://doi.org/10.1007/s11869-022-01252-6>

[9] S. Wei, H. Xi, K. Zhang, Y. Yun and H. Li, "Air Quality Time Series Prediction Optimized by Grey Wolf Algorithm," 2022 3rd International Conference on Information Science,

Parallel and Distributed Systems (ISPDS), Guangzhou, China, 2022, pp. 323-328, doi: 10.1109/ISPDS56360.2022.9874066.

[10] H. Chen, M. Guan and H. Li, "Air Quality Prediction Based on Integrated Dual LSTM Model," in *IEEE Access*, vol. 9, pp. 93285-93297, 2021, doi: 10.1109/ACCESS.2021.3093430.

[11] N. Jin, Y. Zeng, K. Yan and Z. Ji, "Multivariate Air Quality Forecasting With Nested Long Short Term Memory Neural Network," in *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 8514-8522, Dec. 2021, doi: 10.1109/TII.2021.3065425.

[12] Zhou, Xinxing Xu, Jianjun Zeng, Ping Meng, Xiankai. (2019). Air Pollutant Concentration Prediction Based on GRU Method. *Journal of Physics: Conference Series*. 1168. 032058. 10.1088/1742-6596/1168/3/032058.

[13] Y. Wang, S. Zhu and C. Li, "Research on Multistep Time Series Prediction Based on LSTM," 2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE), Xiamen, China, 2019, pp. 1155-1159, doi: 10.1109/EITCE47263.2019.9095044.

APPENDIX

Plagiarism Report:

Similarity Index		Similarity by Source	
14%		Internet Sources	8%
		Publications	8%
		Student Papers	7%
include quoted		include bibliography	
exclude small matches		mode: <input type="text" value="quickview (classic) report"/>	
<input type="button" value="print"/>	<input type="button" value="refresh"/>	<input type="button" value="download"/>	
1% match ("ECAI 2020", IOS Press, 2020) "ECAI 2020", IOS Press, 2020			
1% match (student papers from 15-Mar-2023) Submitted to Liverpool John Moores University on 2023-03-15			
1% match (Hongqian Chen, Mengxi Guan, Hui Li. "Air Quality Prediction Based on Integrated Dual LSTM Model", IEEE Access, 2021) Hongqian Chen, Mengxi Guan, Hui Li. "Air Quality Prediction Based on Integrated Dual LSTM Model", IEEE Access, 2021			
1% match (student papers from 28-Apr-2023) Submitted to University of Hong Kong on 2023-04-28			
1% match (Internet from 14-Jan-2023) http://www.besinner.net/bitstream/handle/106031670346/2020_Teile_Sensoren%20Sensoren_Errollo.pdf?isAllowed=ny&sequence=1			
1% match (Yuan Zhongjie, Wang Shengwei, Wang Ze. "Air quality prediction method based on the CG-LSTM", 2022 5th International Conference on Data Science and Information Technology (DSIT), 2022) Yuan Zhongjie, Wang Shengwei, Wang Ze. "Air quality prediction method based on the CG-LSTM", 2022 5th International Conference on Data Science and Information Technology (DSIT), 2022			
<1% match (student papers from 09-Jun-2023) Submitted to Liverpool John Moores University on 2023-06-09			
<1% match (student papers from 30-May-2023) Submitted to NPP College of Professional Studies Limited on 2023-05-30			
<1% match (Tishya Hanna, A. Anitha. "Forecasting Air Quality Index based on Stacked LSTM", 2022 IEEE 7th International Conference on Recent Advances and Innovations in Engineering (ICRAIE), 2022) Tishya Hanna, A. Anitha. "Forecasting Air Quality Index based on Stacked LSTM", 2022 IEEE 7th International Conference on Recent Advances and Innovations in Engineering (ICRAIE), 2022			
<1% match (student papers from 18-May-2023) Submitted to University of Northumbria at Newcastle on 2023-05-18			
<1% match (Landi Zhou, Ming Chen, Qingjian Ni. "A hybrid Prophet-LSTM Model for Prediction of Air Quality Index", 2020 IEEE Symposium Series on Computational Intelligence (SSCI), 2020) Landi Zhou, Ming Chen, Qingjian Ni. "A hybrid Prophet-LSTM Model for Prediction of Air Quality Index", 2020 IEEE Symposium Series on Computational Intelligence (SSCI), 2020			
<1% match (student papers from 08-Jun-2023) Submitted to UNITEC Institute of Technology on 2023-06-08			
<1% match (student papers from 29-May-2023) Submitted to UNITEC Institute of Technology on 2023-05-29			
<1% match ("VLSI Design and Test", Springer Science and Business Media LLC, 2022) "VLSI Design and Test", Springer Science and Business Media LLC, 2022			
<1% match (Chun Yan Lee, Yi-Ping Phoebe Chen. "Prediction of drug adverse events using deep learning in pharmaceutical discovery", Briefings in Bioinformatics, 2021)			

Fig. 18: Plagiarism Report