

1. INTRODUCTION:

With the development of industry, air pollution has become a serious problem. It is very important to create an air quality prediction model with high accuracy and good performance. Globally, the problem of poor air quality as an environmental risk has grown significantly. As a result, predicting and assessing ambient air quality has grown in importance. In general, the term "air quality" refers to the measurement of clean air in a certain area. It is calculated by measuring several atmospheric pollution indicators. The concentration of air pollutants is estimated using traditional methods, which are computationally intensive. Additionally, these approaches are unable to make sense of the wealth of information at hand. The suggested work offers a deep learning method for quantifying and forecasting ambient air quality in order to address this problem. Long Short-Term Memory (LSTM) is a unique type of structured memory cell that is used in a framework based on recurrent neural networks (RNNs) to do air quality prediction. In this work, we proposed LSTM and GRU (Gated Recurrent Unit) to capture the dependencies in various pollutants and to perform air quality predictions.

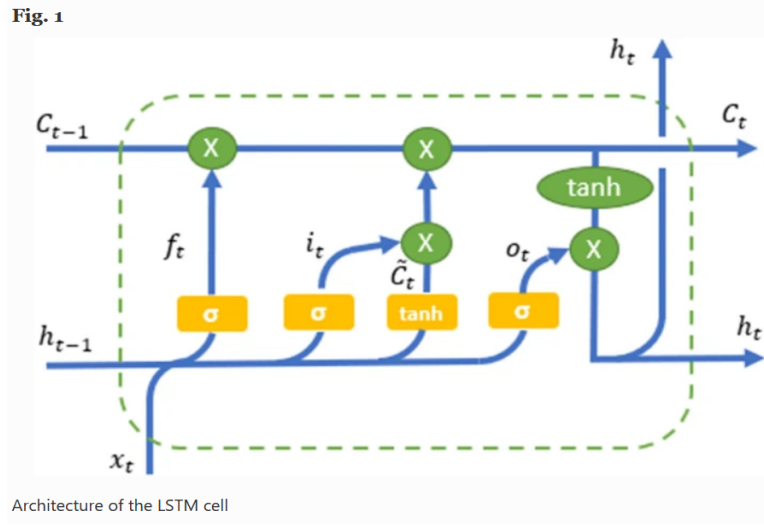
2. SURVEY:

Numerous statistical methods are used to predict the air quality. In this study, we look into the abilities of different deep learning models to predict the concentration of PM_{2.5}. As a result, we choose to employ the LSTM, GRU and DeepAR. Then, we succinctly outline each network:

2.1. LSTM [Long Short-Term Memory]

LSTM is an improved approach to conventional RNN. By including a memory block, LSTM resolves the RNN's vanishing gradient issue. With a constant error carousel (CEC) value of 1, a memory block is a complicated processing unit that has at least one memory cell as well as a few multiplicative gates serving as its input and output. The memory block does not receive any outside signal values, which causes the error value to become active for the duration of the time step. The entire operation of the memory block is under the control of the multiplicative gates. An input gate regulates the flow of input into a memory cell by controlling the activation of the cell. Three gates make up an LSTM: an input gate that decides whether to accept fresh data, a forget gate that eliminates unimportant information, and an output gate that selects the information to be produced. These three gates operate in the 0 to 1 range and are analogical gates based on the sigmoid function.

Fig. Below shows these three sigmoid gates. The cell state is represented by a horizontal line that passes through it.



The LSTM architecture estimates an output $o_t = (o_{t-1}, o_{t-2}, o_{t-3}, \dots, o_{t-T-1}, o_{t-T})$ by updating three multiplicative units (input I output (op) , and forget gate (fr)) on the memory cell with continuous write, read, and reset operations on the memory cell (mc) from time $t=1$ to T .

LSTM formulas are listed below:

$$\text{Inputgate} : i_t = \sigma(W^{(ii)} \bar{x}_t + W^{(ii)} h_{t-1}) \quad (1)$$

$$\text{Forgetgate} : f_t = \sigma(W^{(if)} \bar{x}_t + W^{(if)} h_{t-1}) \quad (2)$$

$$\text{Outputgate} : o_t = \sigma(W^{(io)} \bar{x}_t + W^{(io)} h_{t-1}) \quad (3)$$

$$\text{ProcessInput} : \tilde{C}_t = \tanh(W^{(i\tilde{c})} \bar{x}_t + W^{(i\tilde{c})} h_{t-1}) \quad (4)$$

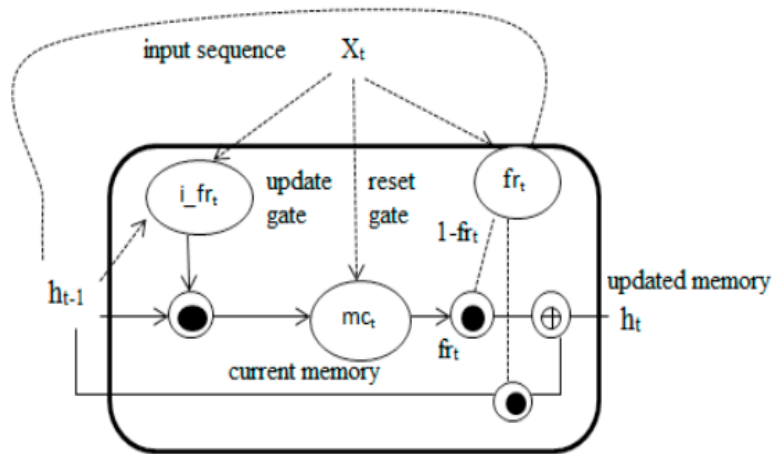
$$\text{Cellupdate} : C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (5)$$

$$\text{Output} : y_t = h_t = o_t + i_t * \tanh(C_t) \quad (6)$$

Since LSTMs are frequently used for sequential analysis, they can be trained to forecast air quality index levels for the upcoming hour or even the upcoming month using the historical data gathered by sensors at different weather stations.

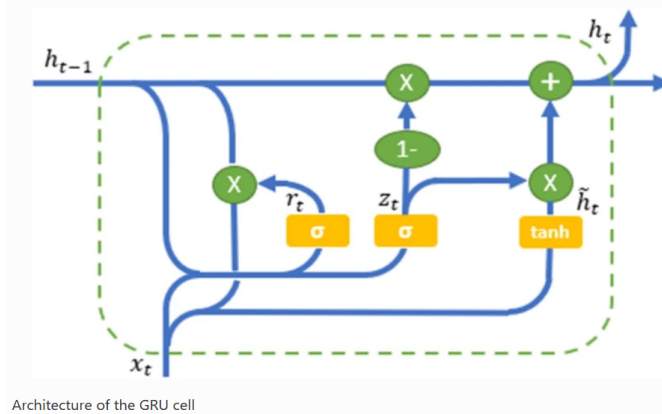
2.2. GRU [Gated Recurrent Unit]:

An addition to the LSTM network is GRU. Update and reset gates make up the system. They all include balancing the data flow within the unit. The GRU receives time series data from AirNet as input. The GRU, or gated recurrent unit, is an improvement over the traditional RNN [33] and is incorporated into RNN. It is comparable to an LSTM unit. The reset and updates gates make up the GRU unit. The GRU architecture is shown in Figure below. While the update gate is used to choose the number of the candidate activation that updates the cell state, the reset gate is intended to erase the previous state between the previous activation and the next candidate activation.



The update gate is used to regulate how much data from hidden states in the past is carried over into the current state. More information about past states is brought in as the update gate value increases. The amount to which the information from earlier stages is discarded is controlled by the reset gate, and the smaller the value of the reset gate, the more it is ignored. As a result, long-term dependencies are accompanied by the activation of update gates, but short-term relationships are typically captured with frequent activation of reset gates.

Fig. 2

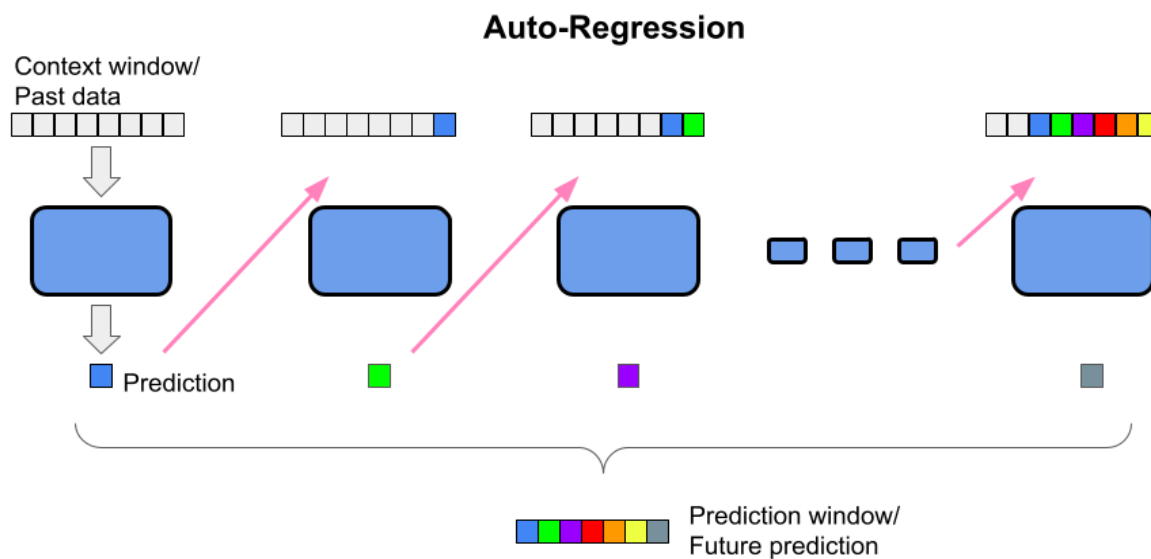


Architecture of the GRU cell

2.3. DeepAR:

DeepAR is a forecasting algorithm used for forecasting scalar one dimensional time-series data. It's a recursive neural network designed by Amazon research group. Classic forecasting methods like ARIMA and ETS can only be trained on individual time-series data and forecast it. However, DeepAR is designed to learn multiple related timeseries and forecast using the combined knowledge of all the related data. When a dataset contains multiple related time series, DeepAR outperforms the traditional ARIMA and ETS algorithms.

The model has multiple tuneable hyperparameters like `context_length`, `prediction_length`, `learning_rate`, `dropout_rate`, `embedding_dimension`, `num_cells` and `num_layers`. The `context_length` decides the number of past records which the model has visibility on. And `prediction_length` decides the number of future records the model can predict. Amazon suggests to keep `prediction_length` less than or equal to `context_length` to ensure model's predictions are close to real values.

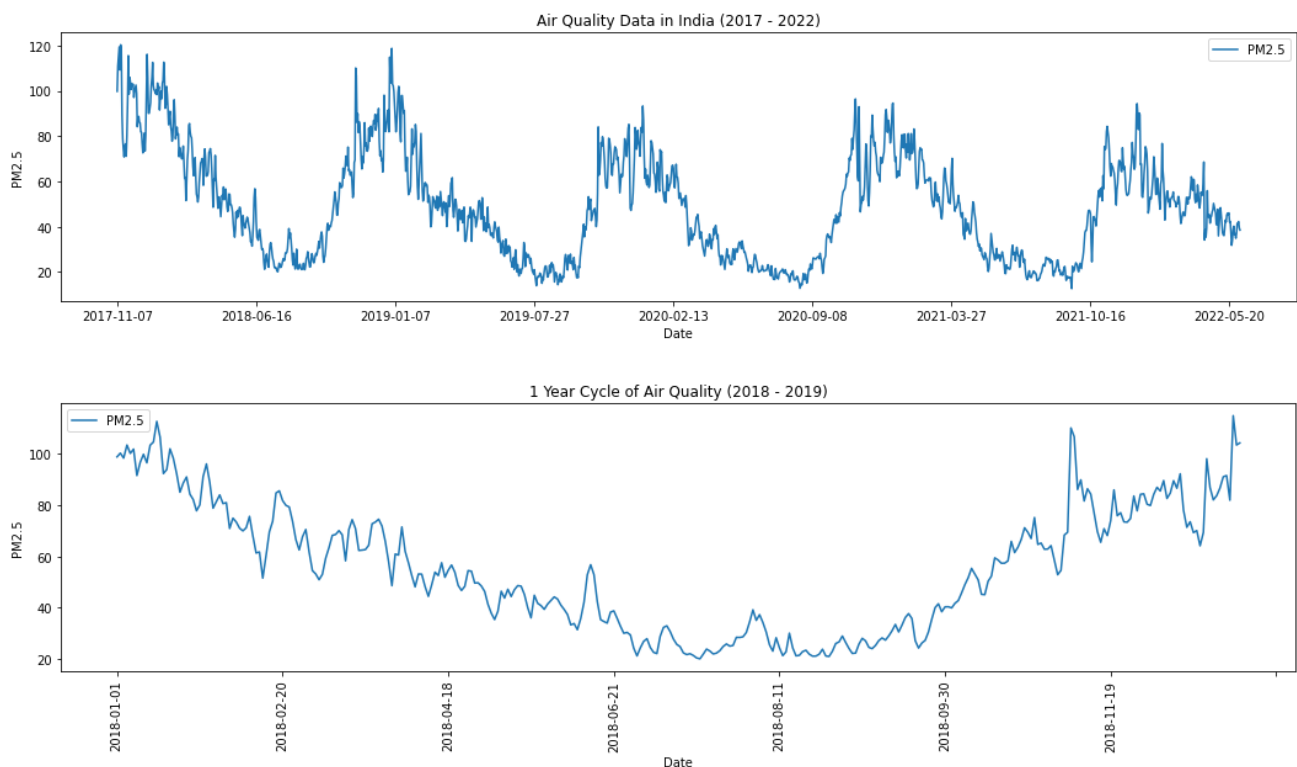


time steps the auto-regressive model can be called to predict the future. An auto-regressor with prediction window 10 can predict up-to 10 days in the future considering each time step accounting for 1 day.

3. IMPLEMENTATION:

3.1. Visualizing of Time Series

When evaluating a time series model's stability over time, rolling analyses are frequently used. An important presumption when using a statistical model to analyse financial time series data is that the model's parameters won't change over time. To illustrate the outcome of the anticipated air quality plot, we use matplotlib. The following graphs show the Air quality of India (2017-2022) and Air quality of 1 year (2017-2018). We have chosen to average the data from hourly basis to daily basis. This illustration gives us an idea of how the patterns in data change over time.



3.2. Prediction using LSTM Model:

It is a sequential model, which is made up of a linear stack of layers. Then we introduced dense, which is the standard deeply coupled neural network layer. It is used to change the output vector's dimensions during backpropagation. Next, we establish our RNN using a regression model. Read the sequential data in and add it to the regressor model to achieve this.

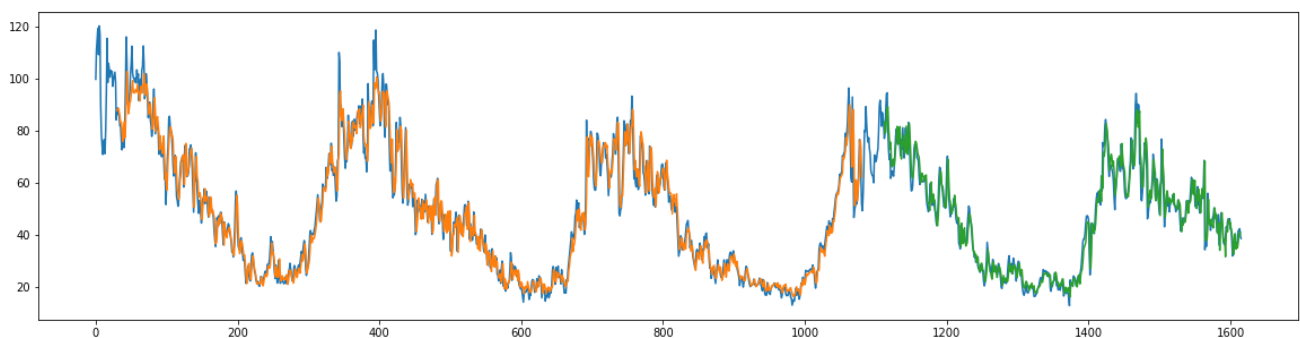
The neural network receives the input and is trained for prediction using random biases and weights. A sequential input layer, three LSTM layers, and a dense output layer with a linear activation function make up the LSTM model. The output value produced by the RNN's output layer is contrasted with the desired value. The backpropagation algorithm reduces error, or the discrepancy between the desired output value and what is actually produced.

Output Value:

The testing scores obtained for LSTM model are as follows:

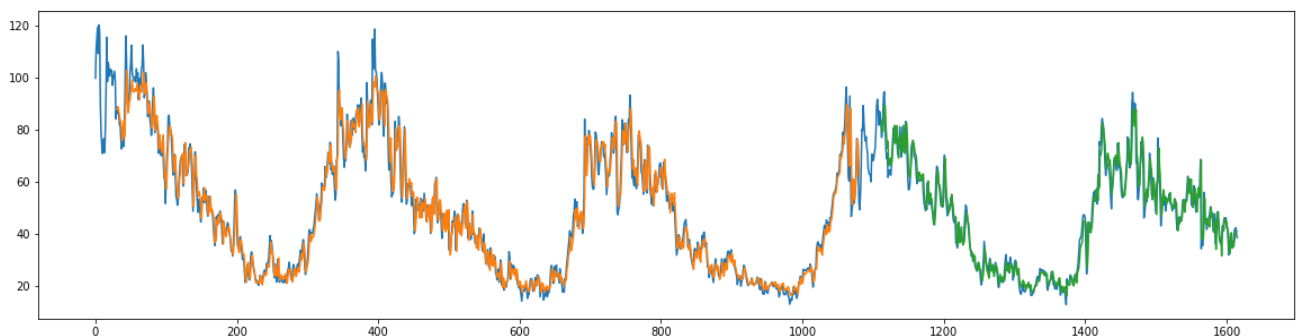
Test Score: 4.82 RMSE
Test Score: 0.93414 R2

The graph obtained below is plotted against models predictions over training data, testing data and actual data.



We also employ auto-regression technique over the existing LSTM model with length of **context window as 100** and length of **prediction window to be 365**. This model takes 100 days data as input and can predict up-to 365 days in the future which is a full 1 year prediction based on past 3 months data. The following graph shows the output of the auto-regression model trying to predict 365 days into the future using past data.

As the context window is 100, the model's predictions are reliable up-to 100 days into the future. i.e. a Prediction window with 100 length is reliable. However further predictions are made based on the previous predictions which will result in higher deviation from the original data.



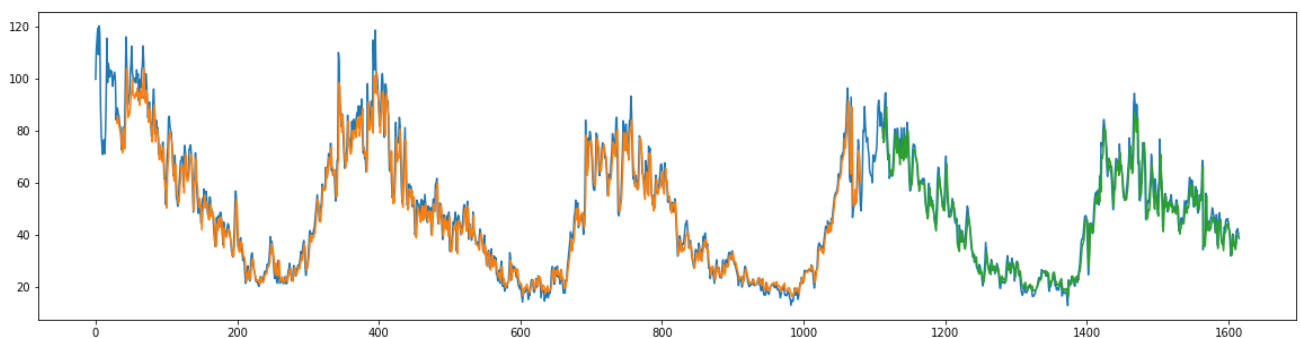
3.3. Prediction using Bi-directional LSTM:

Similarly, we carried out the experiment using bidirectional LSTM model with 1 layer as well as an add on and the outputs turned out to be as follows:

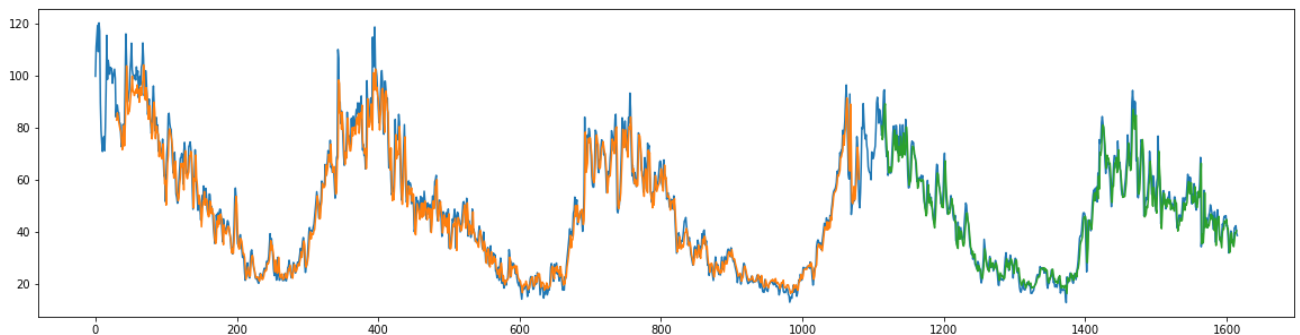
The testing scores and root mean squared errors obtained for this model are:

Test Score: 4.80 RMSE
Test Score: 0.93461 R2

The graph obtained below is plotted against predictions over training and testing data overlapped over original values.



Similarly, we also employed an Auto-regressive approach to the existing Bi-LSTM model and the following graph represents the prediction of 365 days based on the past data.

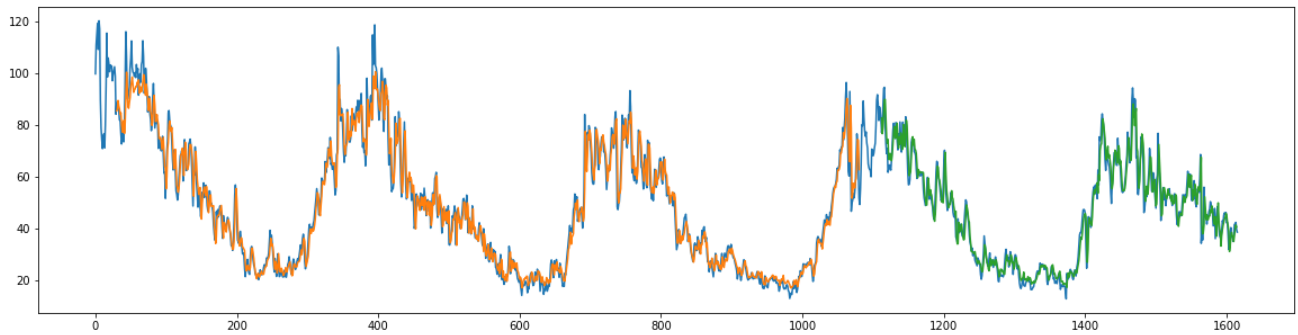


3.4. Prediction using GRU Model:

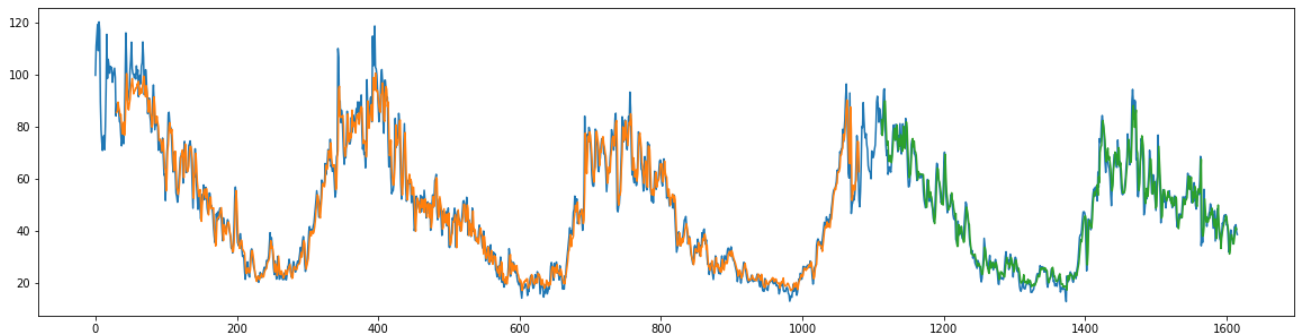
Our experiment consists of usage of GRU for prediction. The test scores obtained for GRU are as follows:

Test Score: 4.80 RMSE
Test Score: 0.93461 R2

The graph obtained below is plotted against predictions over training and testing data overlapped over original values.



Auto-regression output for GRU is as follows.



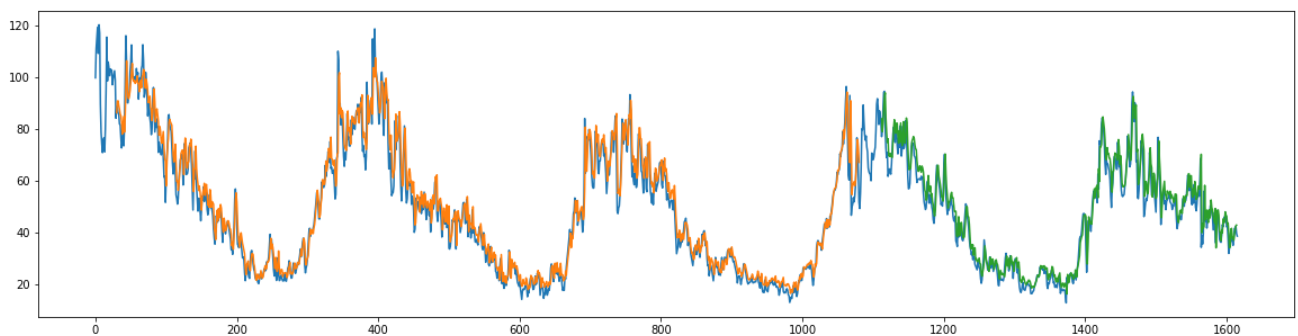
3.5. Prediction using Bi-directional GRU:

Predictions from bi-directional GRU is as follows. The test scores are:

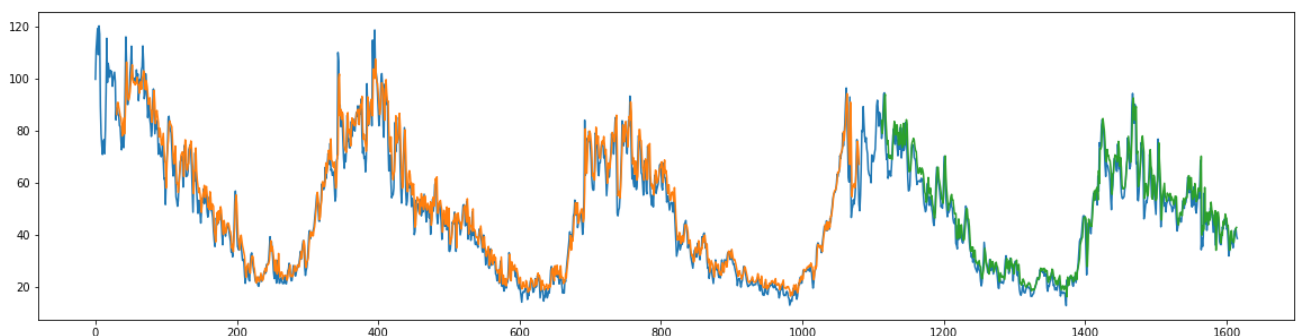
Test Score: 4.80 RMSE

Test Score: 0.93461 R2

The graph obtained below is plotted against predictions over training and testing data overlapped over original values.

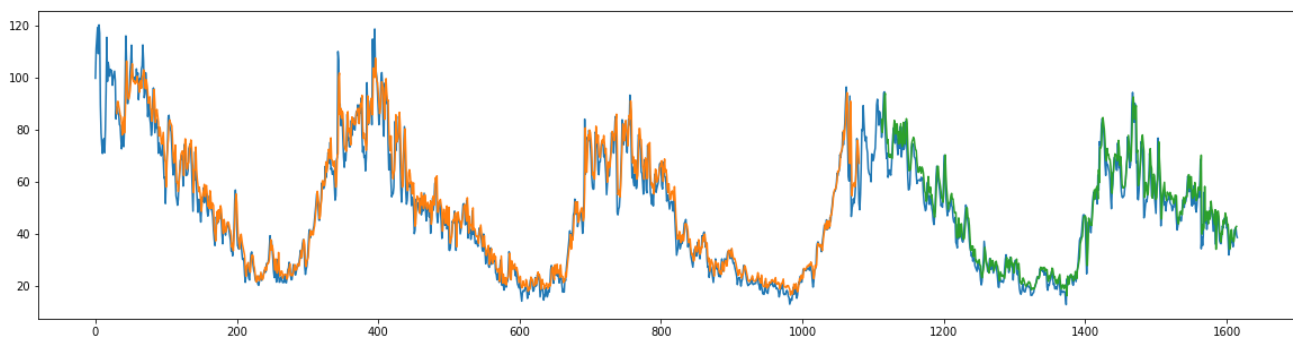


Auto-regression output for GRU is as follows.



3.6. Prediction using DeepAR:

DeepAR algorithm has also been implemented over the data. The experiment was to understand and compare the performance of various models vs the state of the art DeepAR model. The model performed well over the data but the training time was relatively huge and was increasing exponentially with increase in prediction_window. By using a prediction window of 100 days, the model was trained for 10 epochs and 3minutes per epoch. The results of the DeepAR model are provided in the graph below.



4. CONCLUSION:

The R2 values of all the models is in the range of (0.90 - 0.92). The highest score has been reached by the bi-directional GRU model being 0.9175. According to the findings, the Bi-GRU network performs marginally better than the LSTM network. Additionally, bidirectional GRU has a reduced error value when compared to any other model, which suggests that its adoption can enhance prediction accuracy. This is because the bidirectional GRU processes the time series both chronologically and anti chronologically, capturing patterns that one-direction GRUs could overlook and enhancing the ability of time series to learn features.

5. FUTURE WORK:

In the future, we can improve our model even further and assess it using a larger dataset. We can also use the programme to forecast the concentration of other pollutants. By using techniques like Convolution Neural Network (CNN), the analysis can be expanded further in order to detect the unequal changes occurring in the air pollution data. The relationship between several characteristics can also be evaluated, allowing us to determine whether a concealed parameter will correlate the performance of features that seem to perform differently from the initial glimpse. As both LSTM and GRU demonstrate their significance in prediction, there may as well be a chance that their combined model will be more effective than LSTM and GRU used alone.

6. REFERENCES:

- 1) Wang, Jingyang, Jiazheng Li, Xiaoxiao Wang, Jue Wang, and Min Huang. "Air quality prediction using CT-LSTM." *Neural Computing and Applications* 33, no. 10 (2021): 4779-4792.
- 2) Tiwari, Animesh, Rishabh Gupta, and Rohitash Chandra. "Delhi air quality prediction using LSTM deep learning models with a focus on COVID-19 lockdown." *arXiv preprint arXiv:2102.10551* (2021).
- 3) Athira, V., P. Geetha, Rab Vinayakumar, and K. P. Soman. "Deepairnet: Applying recurrent networks for air quality prediction." *Procedia computer science* 132 (2018): 1394-1403.
- 4) G. Dantas, B. Siciliano, B. B. Franc¸a, C. M. da Silva, and G. Arbilla, "The impact of COVID-19 partial lockdown on the air quality of the city of rio de janeiro, brazil," *Science of the Total Environment*, vol. 729, p. 139085, 2020
- 5) Zhang, Luo, Peng Liu, Lei Zhao, Guizhou Wang, Wangfeng Zhang, and Jianbo Liu. "Air quality predictions with a semi-supervised bidirectional LSTM neural network." *Atmospheric Pollution Research* 12, no. 1 (2021): 328-339.
- 6) Song, Xijuan, Jijiang Huang, and Dawei Song. "Air quality prediction based on LSTM-Kalman model." In *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, pp. 695-699. IEEE, 2019.
- 7) Zhou, Xinxing, Jianjun Xu, Ping Zeng, and Xiankai Meng. "Air pollutant concentration prediction based on GRU method." In *Journal of Physics: Conference Series*, vol. 1168, no. 3, p. 032058. IOP Publishing, 2019.
- 8) Ahn, Jaehyun, Dongil Shin, Kyuho Kim, and Jihoon Yang. "Indoor air quality analysis using deep learning with sensor data." *Sensors* 17, no. 11 (2017): 2476.
- 9) Bekkar, Abdellatif, Badr Hssina, Samira Douzi, and Khadija Douzi. "Air-pollution prediction in smart city, deep learning approach." *Journal of big Data* 8, no. 1 (2021): 1-21.
- 10) Du, Shengdong, Tianrui Li, Yan Yang, and Shi-Jinn Horng. "Deep air quality forecasting using hybrid deep learning framework." *IEEE Transactions on Knowledge and Data Engineering* 33, no. 6 (2019): 2412-2424.