**An online Air Pollution Forecasting system using LSTM and GRU**

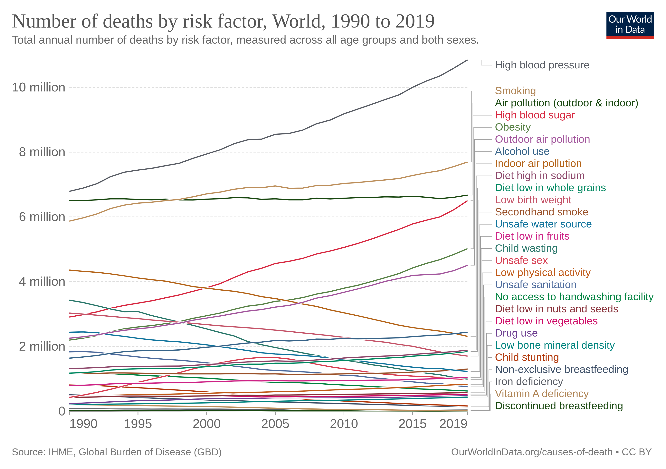
***Abstract--* With the development of the industry in the last few years. We are facing an issue related to air quality. We are not getting the proper air quality because of the pollution out there in the environment due to harmful gases from the industry. It will affect human health and it can cause a serious issues related to the lungs. Air pollution can cause by different ways in today's world like the CO2 released from the car and some harmful chemicals which are released in the air by the industry. Air pollution can spread with the flow of air means wind direction and speed. To overcome this real-time problem there is much research going on nowadays that can forecast air pollution but it required a lot of computational power. Our approach in this research is that we are going to forecast air pollution using the very famous deep learning technique Recurrent Neural Networks (RNN) based framework with special structure memory cell known as Long Short term memory (LSTM) and Gated Recurrent Unit (GRU). We can easily forecast air pollution using this easily by just providing some last day’s data to the model. It will forecast the next 24 hours of data by just providing the last 15 days of air pollution of data.**

1. INTRODUCTION

According to the WHO (World Health Organization), air pollution is the contamination of the indoor or outdoor environment by any chemical, physical or biological agent that modified the natural characteristics of the atmosphere. Air pollution can be divided into 2 parts indoor pollution from households and outdoor pollution from vehicles and industry. Air pollution can be felt by Household combustion, motor vehicles, industrial facilities, and forest fires are common resources of air pollution. WHO data shows that almost all the global population (90%) breathes air that exceeds [WHO guideline limits](https://www.who.int/publications/i/item/9789240034228). Every 9 out of 10 people lives where air quality exceeds WHO guidelines. The World Health Organization (WHO) reported that air pollution causes 4.2 million premature deaths per year in cities and rural areas around the world. Air pollution in the cities and rural areas causes some dangerous diseases like stroke, heart disease, lung cancer, and acute and chronic respiratory diseases. Around the globe around 2.6 billion, people are exposed to dangerous levels of household air pollution. This is the data from WHO.

Air pollution forecasting techniques are being rapidly advanced and measuring pollution increase. Traditional approaches use some mathematical and statistical techniques[1]. This conventional forecasting model takes a lot of computational power to forecast the data. With recent advancements in technology, we come up with Deep Learning which is very good for solving real-time problems in various domains like computer vision, Natural Language Processing, and many more. With the promising results obtained by the Deep Learning model, we can adapt this to forecast air pollution[1].

As you can see in the world Air pollution leads the third largest cause of death[[link](https://ourworldindata.org/air-pollution#how-are-death-rates-from-air-pollution-changing)].



With the help of a deep learning approach, we can use the RNN (Recurrent Neural Networks) based framework which is LSTM (Logn short term memory) and Gated Recurrent Unit (GRU) with a special kind of memory cell attached to it. We have created a model with the help of the input layer (For providing the data to the model), 2 hidden layers (To process and learn about the pattern), and output layers (For output what the model has generated using the hidden layer). We can forecast the data with the help of providing the data of the last 15 days of data and we can forecast the next 24 hours’ data which will be accurate according to the last 15 days of data.

1. LITERATURE SURVEY

Deep Learning approaches have emerged as powerful solutions to mitigate these limitations over conventional methods [1]. The most popular Deep Learning techniques are Multi-Layer Perceptron (MLP), Deep Belief Network (DBN), Convolutional Neural Networks (CNN), Recurrent Neural Network (RNN), and Auto Encoder (AE). A particular RNN-based model for predicting air quality has drawn much attention in recent times [1].

Many researchers are working on this problem of air pollution forecasting nowadays. Mostly they are focusing on the LSTM (Long short term memory) or GRU (Gated Recurrent Unit). But in this research, we are going to combine both of the famous models of Recurrent Networks[4]. In many fields, these 2 models are giving their best to provide the solution to the problem.

1. METHODOLOGY

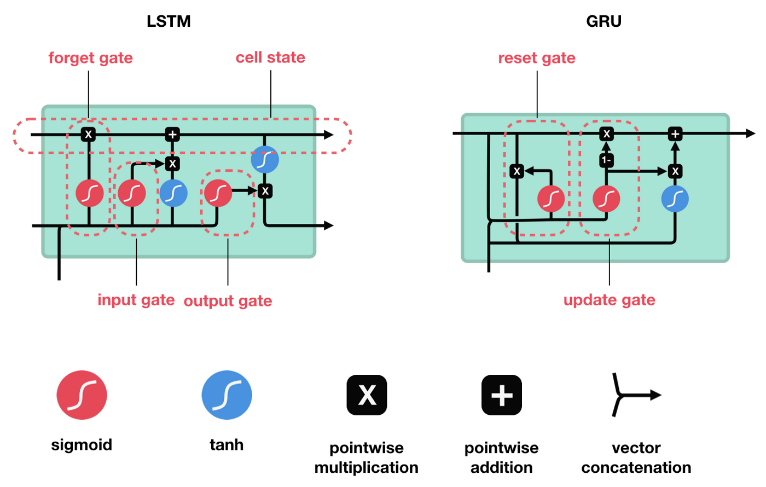
In this paper, we are going to use the most popular framework of Deep Learning which is LSTM (Long Short term memory) and GRU (Gated Recurrent Unit). As we all know LSTM which has the special ability for storing the previous execution data and store in the memory and can be used for predicting data. Recurrent Neural networks suffer from short-term memory. If a sequence is long enough, they will have a hard time carrying information from earlier time steps to later ones.

LSTM and GRU were created as the solution to short-term memory. They have internal mechanisms called gates that can regulate the flow of information. These gates can learn which data in a sequence is important to keep or throw away. The LSTM has a similar control flow as the recurrent neural network. It processes data passing on information as it propagates forwards. Every LSTM memory cell has the following Sigmoid and Tanh Activation function and three gates which are Input Gate, Forget Gate, and Output Gate.

The sigmoid activation function converts between 0 and 1. That is helpful to update or forget data because any number multiplied by ‘0’ will convert to 0, which can consider forgetting the data from the memory cell. And if the output is 1 then the value should be considered as the important data and kept that data in the memory cell. As we can see in the figure that the graph is from 0 to 1 and convert the values

between 0 and 1.

As similar to the sigmoid Activation function even tanh is also playing a major role in the LSTM memory cell. Tanh will squish the values between -1 and 1 and this is useful in the Input gate and convert the values between -1 and 1 and multiply with the existing data of that Input gate.



As we can see in the figure that in LSTM we have 3 gates and in GRU we only have 2 gates. Both the LSTM and GRU are using the sigmoid activation function for the in there gate architecture. If the outcome for the sigmoid is 0 then forget the data and if the outcome is 1 then put the data and go further step. In Forget gate, it will decide whether the data which we are providing for the processing are important or not based on the output of the sigmoid function. Same for the GRU reset gate will define whether the cell state should reset or not based on the sigmoid activation function. If the outcome is 0 then reset the cell state or if 1 then keeps going on.

In LSTM we have 3 gates namely forget gate , Input gate and output gate. Firstly when data comes from the previous cell state it comes into the forget gate here we are defining that weather we have to put the data or forgot the data. We will do that with the help of the sigmoid activation function (The output of sigmoid function will 0 or 1) after generating the output based on the input which we have received from the previous cell state. We will multiply that data to the current cell state So if the output will be 0 then we will forgot the data or else if 1 then we will keep data and store into the current cell state. Here the task of forgot gate will over. Now the work for the input gate started and in that we will give the data from the previous cell state. Here firstly we will generate the output from the sigmoid activation function and along with that we will generate the output for the tanh (It will generate the output between -1 and +1) activation function. Then we will multiply it and add it to the current cell state. From the input gate we can understand that whether the data which we providing to the current cell state is important or not. Here the work of input gate will over. After that it comes the output gate in which we will first generate the output for the tanh activation function with the help of current cell state and then generate the output for the sigmoid of the previous cell state and then we will do product of that and send it to the next cell state. And then for the next cell state the process will continue like this.

In GRU we only have 2 gates which are reset gate and update gate. The reset gate is similar to the forgot gate in the LSTM , it will justify whether the data from the previous cell state we have to keep it or reset that cell state. After that it comes the update gate. In update gate we will update the cell state with the help of sigmoid activation function and add the output of that with -1 and multiply it to the cell state. For the output of the Memory cell it will multiply with tanh activation function and then multiply with the update gate output and add it to the current cell state

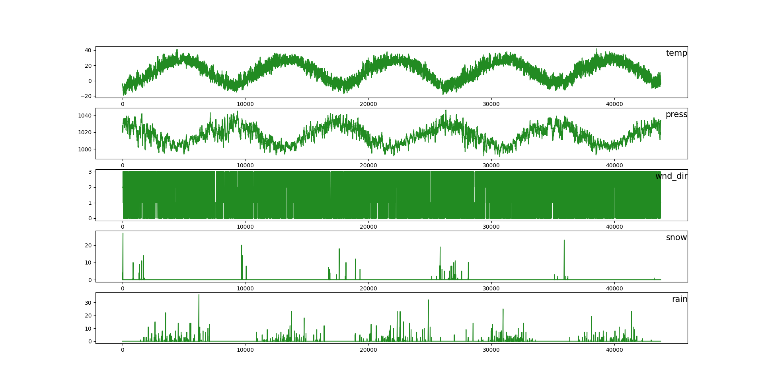
We have used both the LSTM and GRU models for forecasting data. In our model, you have to give the last 15 days of data with the parameters of { date , pollution , dew , wind\_dir , wind\_spd , snow , rain , pollution } based on that we can forecast the next 24 hours of data which is pollution. We have to build a model to predict the next 24 hours of data. In this, the date is playing the most important role in forecasting data.

For the implementation of this system we have to follow these steps :

1. Data Pre-processing
2. Creating a model
3. Saving a model
4. Generate output

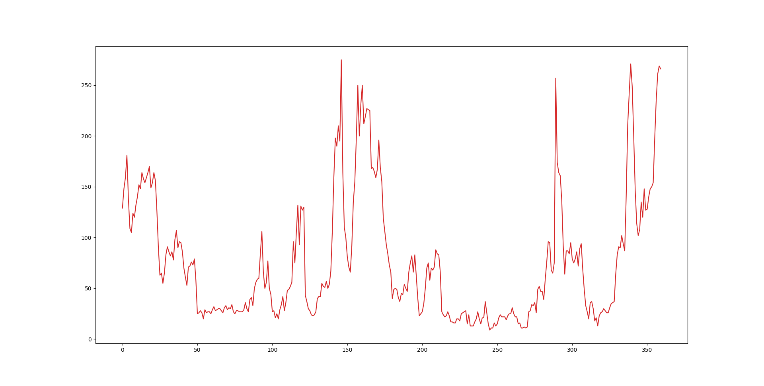
1.Data Pre-Processing :

For Data Pre-processing firstly we have to understand the data which we have. To understand the data we can use the EDA. Firstly if in our data set there is a requirement for the Encoding of data we have to do that we can do that with the help of a Label Encoder or One Hot Encoder. After that, we have to normalize the data so our model can learn from that without getting more confused.

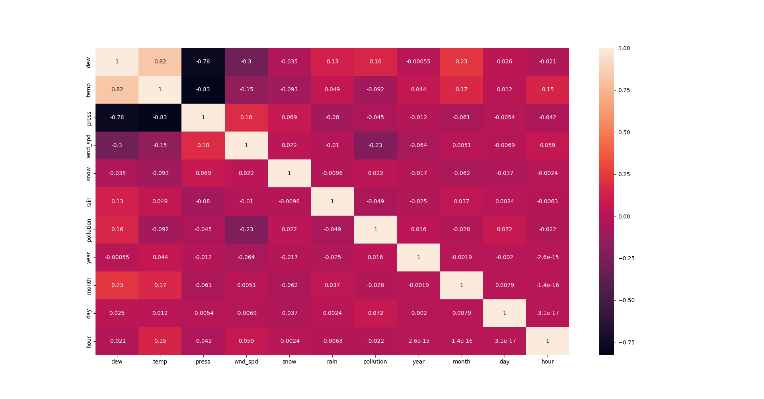
For Label Encoding process of the data the data should be processed and generate the one unique number for the each and every unique data in the column of that dataset. But in the case of One Hot Encoder the for every unique data in the column It will generate one number and rest of them assign to the same values. 

This graph represent the input variable data from the dataset and visualize it. It will easy to understand the data from which we can organize the data.

This below graph represent the data for the output variable for the last 15 days of data with respect to every hour which mean it shows the data of 360 hours.

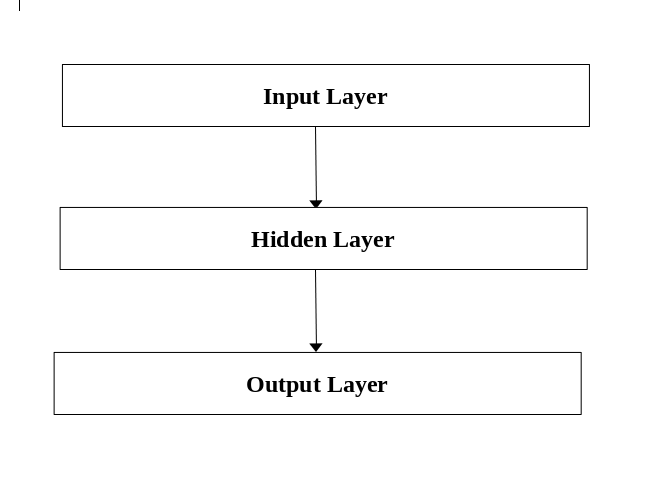


For selecting the features which are appropriate for the model we have to understand the co-relation between the data. From that we can select the features which are useful to us. If we will select the features more which are highly co-related to the each-other then our model will come in the state of overfitting. Which means our data will work more accurate with the training data but when it comes to the testing data it will not predict accurate.



This is the Heat map co-relation of the entire dataset features.

2. Creating a model

For creating the model we have used the LSTM (Long short term memory) like the normal Deep learning architecture it has mainly three layers Input Layer, Hidden Layer, and Output Layer. In the input layer, we have created the input layer of the size of [360(no. of hours),11(no. of features)] with the activation function of relu. After that, we have added another GRU layer of 50 neurons , Dropout Layer , LSTM layer in it with another layer of Dense layer for the output layer and activation function of linear. For this, we have used loss functions of mse(Mean Standard Error), optimizer of Adam Optimizer, and metrics of accuracy. For training the model we have used the epoch of 100 and step\_per\_epoch is 25.

Our model architecture would look like this :

|  |  |  |
| --- | --- | --- |
| Layer (Type) | Output shape | Param # |
| Lstm (LSTM) | (None , 360 , 50) | 12400 |
| Gru (GRU) | (None , 360 , 50) | 15300 |
| Dropout (Dropout) | (None , 360 , 50) | 0 |
| Lstm\_1 (LSTM) | (None , 50) | 20200 |
| Dropout\_1 (Dropout) | (None , 50) | 0 |
| Dense (Dense) | (None , 24) | 1224 |
| Activation (Activation) | (None , 24) | 0 |

3. Saving the model

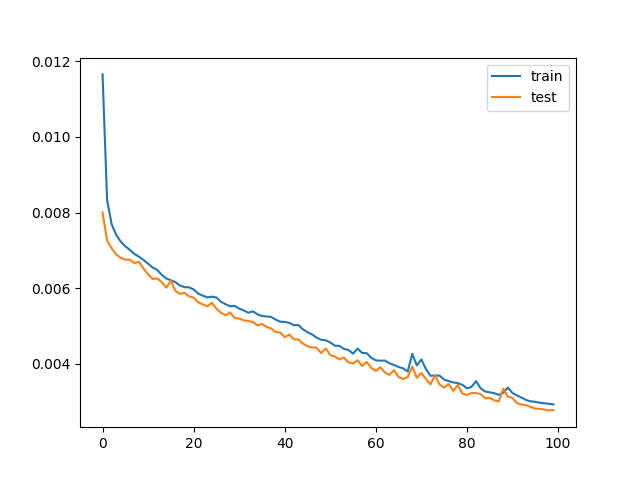
The creation of model is time consuming process. To avoid that process to do it every time we will save that model for the future use. In future if we have to use that model we can use that easily by just calling that model to our application. By the help of saving model we can use that model in any kind of application like web application , we can create a API from that and many more usecase.

To save the model we have used inbuild method of tensorflow which is “model.save(“Enter the name for the model to save”)”.

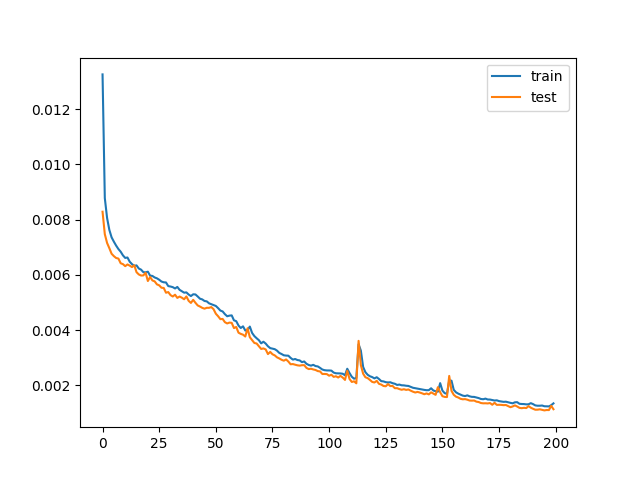
4. To Generate the output

The output which will generate from this model are next 24 hours of air pollution data. We just have to provide the input of the last 15 days of data with respect to the every hour which is in total 360 hours of data. We have to give last 360 hours of data and it will predict the next 24 hours of data.

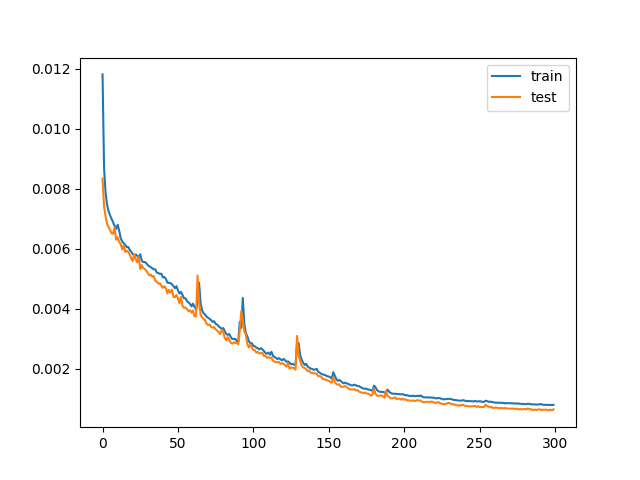
1. RESULT

We have created the model for 3 different epoch to get to know which model is performing the best. For that we have created the model for 100 epoch , 200 epoch and 300 epoch respectively.

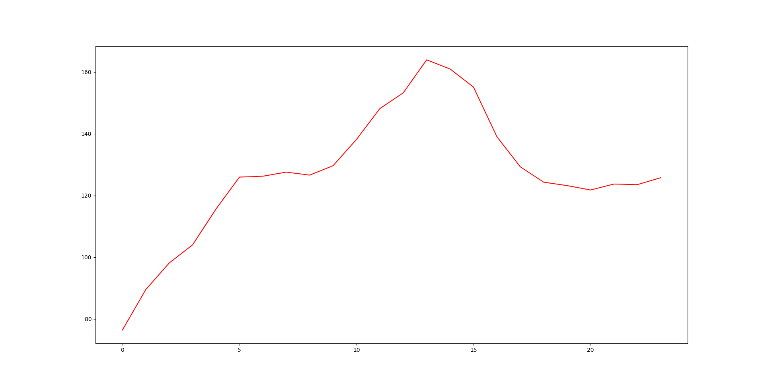
This is the graph for the output of 100 epoch with respect to the validation loss and loss. From the above graph we can say that if we consider this as the model then our model would not perform well and can generate the wrong output also. Here we can see the variations in the test data and train data.



This is the graph for the output of 200 epoch with respect to the validation loss and loss. From the above graph we can consider that our model has performed well in this by seeing we can say that there is no more variations in the data of training data and testing data.



This is the graph for the output of 300 epoch with respect to the validation loss and loss. From the above figure we can consider that if we are using the 300 epoch for creating a model we are getting a good accuracy but if we will see closely then the both 200 and 300 epoch graphs are same. But in the 300 epoch graph at the end the data are getting variations in that. So we will consider 200 epoch model as our final model and we will save that for the further predictions of data.

After that for forecasting the data we have to give last 360 hours of data which is 15 days of data and give it to the generate model which will predict next 24 hours of data.

This is the graph for the next 24 hours of data which our model has predicted.

1. CONCLUSION

The motive of this paper is to determine an efficient forecasting model for the hourly concentration level of air pollution. With the help of this model we can forecast the air pollution and we can see when the air pollution is going to be high. The result of work carried out supports the idea of a deep learning-based technique for forecasting air quality achieving promising performance. This work will take the input of the data for the last 15 days of data and can predict the next 24 hours of data. The result that the model has generated we can say that it is accurate based on the last 15 days of data but we cannot say anything about the environment.

1. REFERENCE