WINDBOT

PROJECT REPORT

**Index**

1. **INTRODUCTION**
   1. OVERVIEW
   2. PURPOSE
2. **LITERATURE SURVEY**
   1. EXISTING PROBLEM
   2. PROPOSED SOLUTION
3. **THEORETICAL ANALYSIS**

BLOCK DIAGRAM

1. **Advanced Working Explanation of the LSTM model**

**5. Technical Details regarding the model**

**6. Working of Node-red backend**

**7. FLOW VIEW**

**8. CHALLENGES FACED**

**9. ADVANTAGES**

**10. DISADVANTAGES**

**11. APPLICATIONS**

**12.CONCLUSION**

**13. BIBLIOGRAPHY**

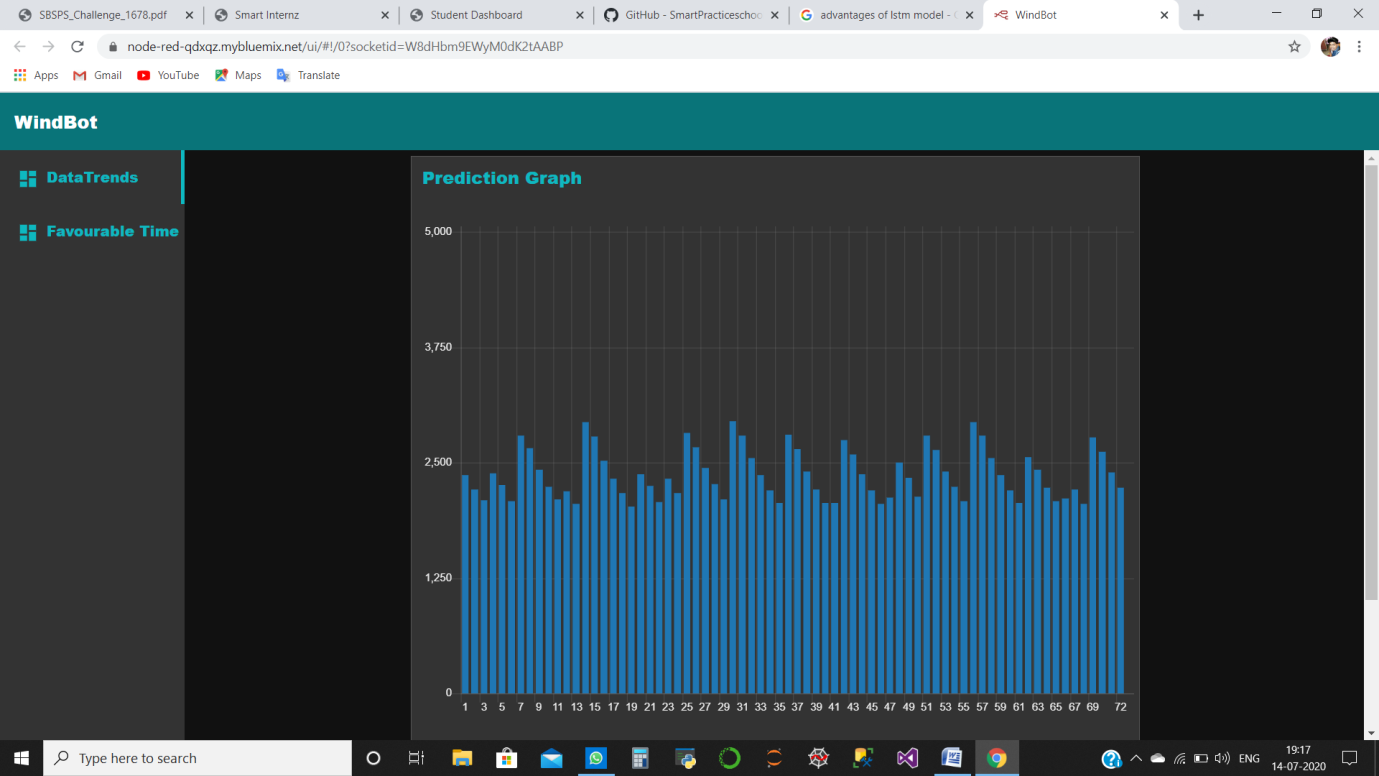
**Introduction**

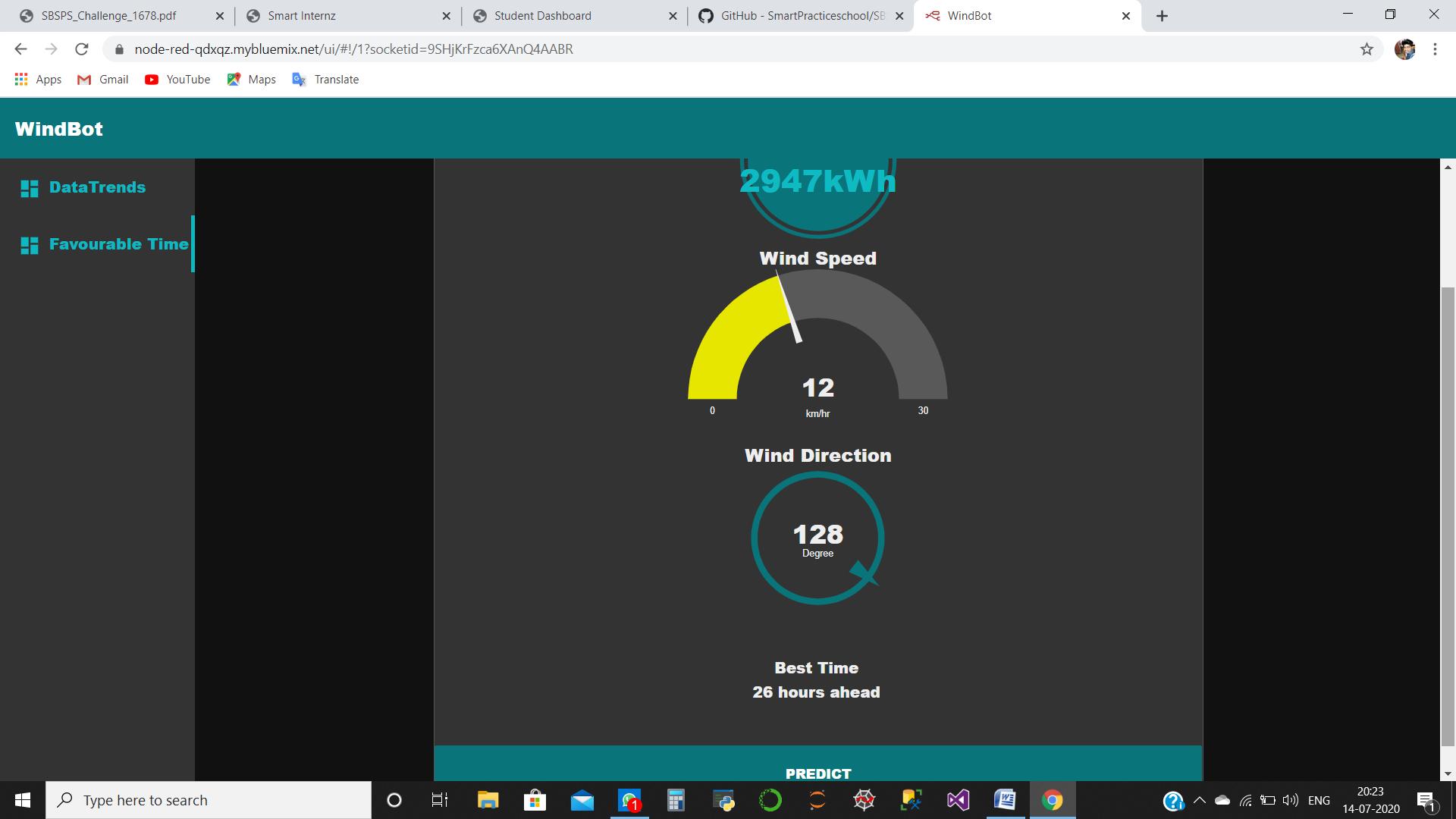
**1.1 Overview**

 Onshore wind is an inexpensive source of electric power, competitive with or in many places cheaper than coal or gas plants.[[2](https://en.wikipedia.org/wiki/Wind_power#cite_note-Lazard2018-2)In 2018, global wind power capacity grew 9.6% to 591 GW[[17]](https://en.wikipedia.org/wiki/Wind_power#cite_note-17) and yearly wind energy production grew 10%, reaching 4.8% of worldwide electric power usage, The energy output of a wind farm is highly dependent on the weather conditions present at its site. If the output can be predicted more accurately, energy suppliers can coordinate the collaborative production of different energy sources more efficiently to avoid costly overproduction. Here we have used the concepts of data science and deep learning algorithm to predict the energy output of wind turbines. To do this, we have followed the following steps:-

* Building an LSTM deep learning model in IBM Watson Studio.
* Creating a Node-Red Server for storing the inputs for our model.
* Building a User Interface.
* Hosting the app on IBM Cloud

**1.2 Purpose**

The app demonstrates two options:-



* **Data Trends**:- this contains a bar chart to show **energy output** from **1 hour to 72 hours.**
* **Favourable Time:-** this shows **Predicted energy output (inkWh),Wind Speed(in km/h)**,**Wind Direction(in degree)** and **Best Time(in h).**
* **Predict(in Favourable time):-**this button is used to **predict the output energy.**

**LITERATURE SURVEY**

**2.1 EXISTING PROBLEM**

We had to develop a time series model to predict the power output of wind farms based on the weather condition in the site (1Hr prediction to 72Hrs. prediction). Further, we had to build an application to recommend the Power Grid the best time to utilize the energy from wind farms.

We used different linear models like LinearRegression(),Lasso(),ElasticNet()and RidgeRegression(),

We even tried RandomForestRegressor(), though R Squared score was within **0.964 to 0.971.** Such R squared score is good but the loss function like mean absolute error and mean square error will be a bit high, causing an error of about **450** between predicted and actual energy output.

**2.2 Proposed Solution**

We switch to LSTM deep learning algorithm. The main reason to switch to this algorithm was to get low loss functions, and further, we also wanted to predict weather conditions too at the best time.

**THEORETICAL ANALYSIS**

**BLOCK DIAGRAM**







**IoT DEVICE**: It will collect Wind Speed and Wind Direction. We are using

random number as we could not afford real

**IOT SERVER**: Responsible for data processing.

**BUFFER**: Stores the data from IoT to be used as input to the model.

**CSV**: It will contain a part of formated data to be sent to the model.

**MODEL**: it is the deployed LSTM model.

**Advanced Working Explanation of the LSTM model**

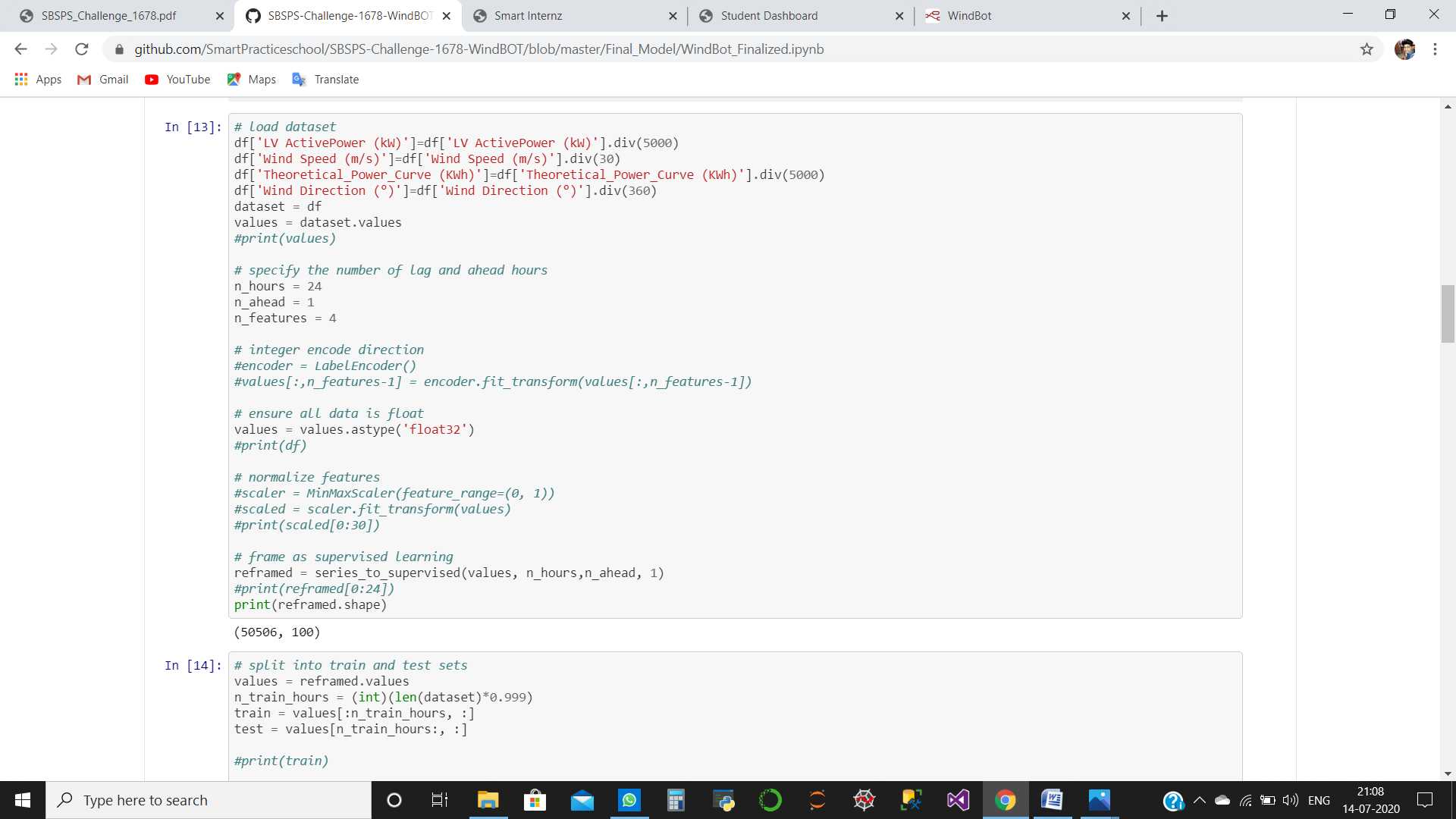
On opening our App. It will automatically detect the time and send it to the node-red server The server will fetch the last 24 hours data r(t-24) to r(t) to the deployed model. The deployed model returns the predicted energy output of the next hour.

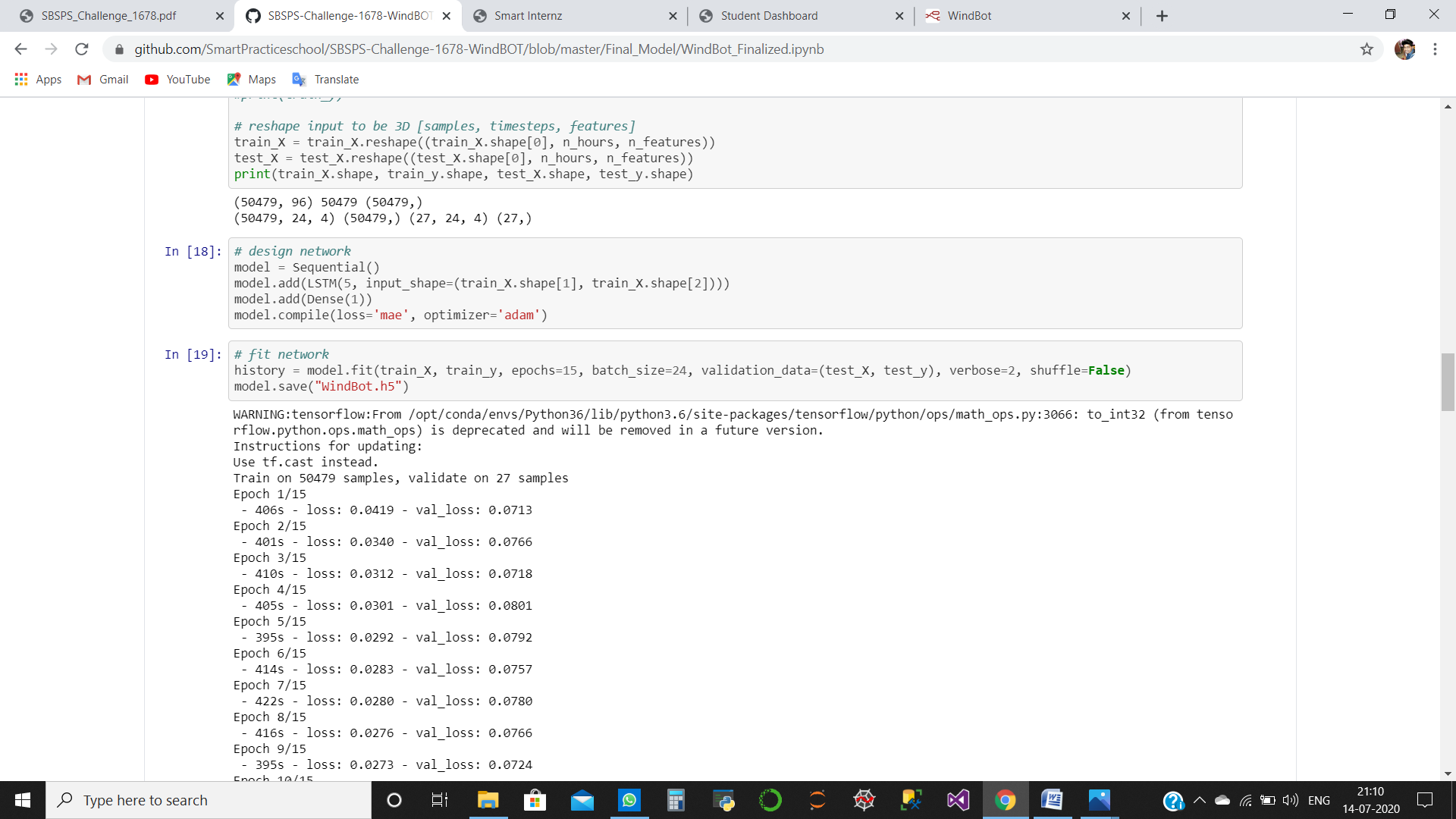
In the next phase, our app will get the wind direction and speed from the database for the next hour. Then it will combine predicted energy along the above data to generate a temporary row p(t+1).

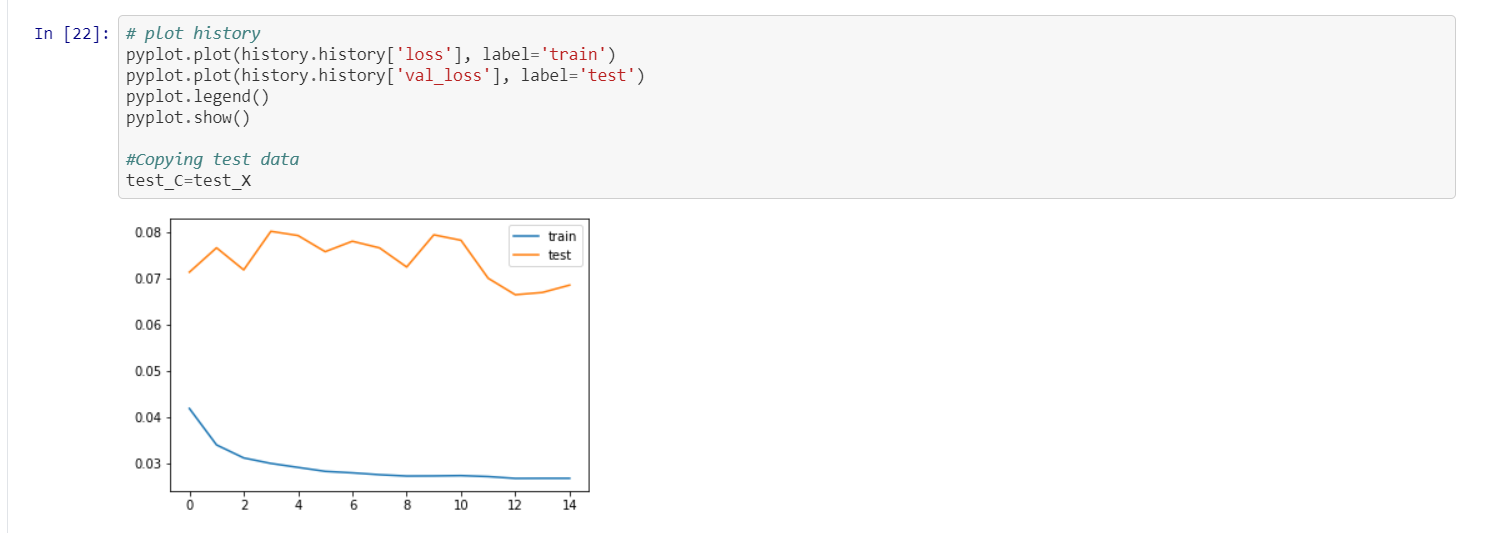
Then we will again send it r(t-23) to r(t) along with p(t+1) to predict energy output of p(t+2). It will continue until we get all p(t+72) rows. We will calculate the best hour to use energy.

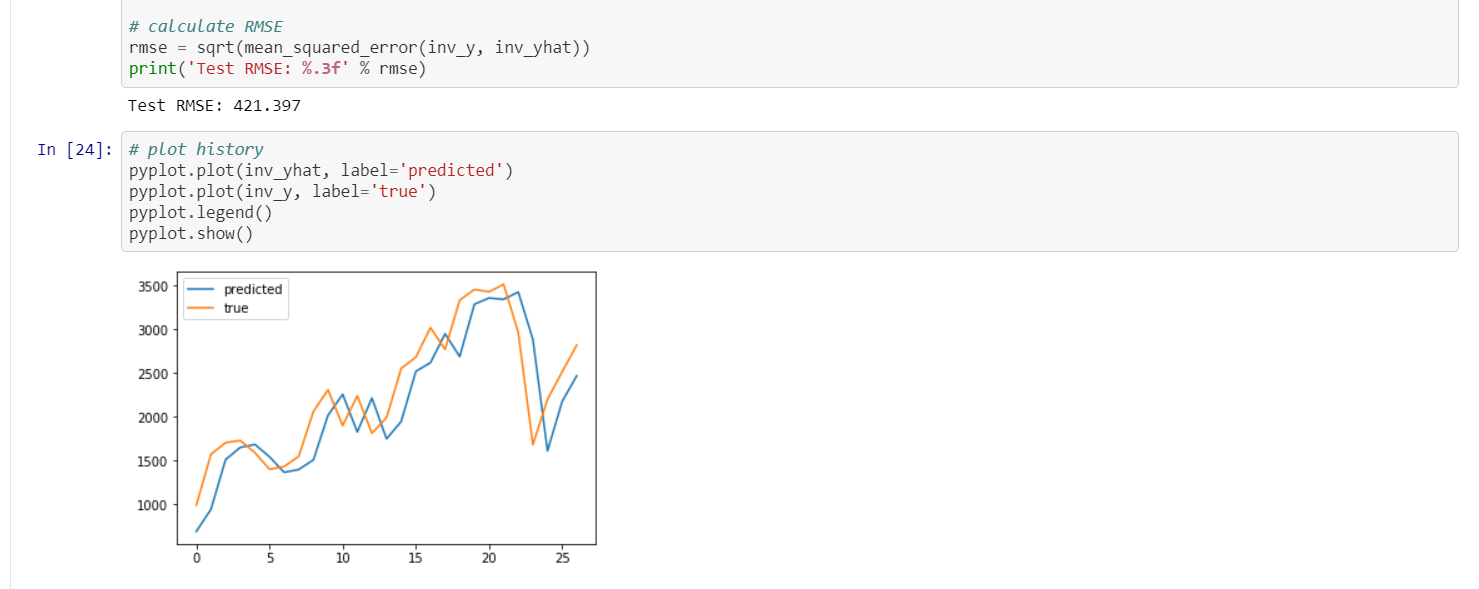
Finally, Node-Red Server will return all the data required to plot a graph of energy prediction for the next 72 hours. It will also display the best hour to use energy.

**Technical Details regarding the model**

****

****





**Working of Node-red backend**

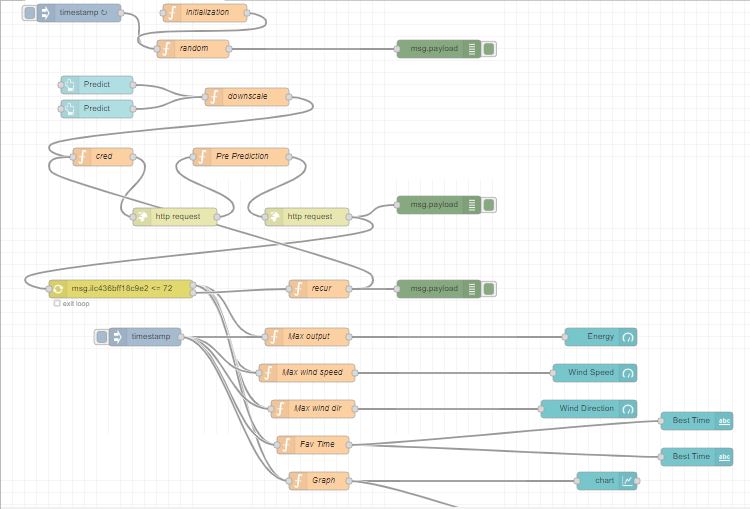
We initialize some global variables which can be used any time in our code.

We record the wind speed, wind direction, actual power output, and theoretical power output of the last 24 hours. We could have done so using the IoT Device but the option for its customization was not available for us. So, we analyzed the dataset and generated random input data based on the most common range given in the provided dataset. The data will keep getting updated every hour and we will have the data of the last 24 hours which we will use for prediction.

Then we downscale our data and supply it to the trained machine learning model. We obtain the data for the next hour and update our input data with it and supply it again to the trained model. Thus we continue this process 72 times and obtain the predictions for the next 72 hours.

Then we upscale our output and supply it to the frontend and display the output graph, maximum output, the wind speed, and wind direction at that output. We will also display the time at which we will obtain the maximum output power.

**Flow-View**

****

**CHALLENGES FACED**

* **Tensorflow version error**:-this was the first challenge that we faced. We found that the version of TensorFlow which IBM cloud was offering was old enough for LSTM models. To overcome this we update TensorFlow to the latest version.
* **Multi-Output not supported**:- as mentioned above our model can also predict wind speed and direction, for this, we had created a model that can give multi-output(energy, speed, direction) at the same time. We found that the IBM cloud does not allow its free users to deploy a multi-output model so we have to switch to single-output(energy), and other outputs were generated randomly. However, if we can easily replace our model with a multi prediction model if we are granted permission.

**Advantages**

Windbot has the following advantages:-

* **High accuracy**:- accuracy of **95.17**% gives a very close value of predicted energy to actual energy.
* It is **very user friendly**, as the app automatically predicts as soon as it is opened.
* It can also **predict Wind Speed and Wind direction** at the best time.

**Disadvantages**

Windbot has following disadvantages:-

* Reprediction takes a long time.
* We need an IoT device to capture inputs, without which the app will not work.

**Applications**

**Renewable power is**[**booming**](https://www.nrdc.org/revolution-now)**, as innovation brings down costs and starts to deliver on the promise of a**[**clean energy future**](https://www.nrdc.org/issues/clean-energy)**. This means that renewables are increasingly displacing “**[**dirty**](https://www.nrdc.org/issues/dirty-energy)**” fossil fuels in the power sector, offering the benefit of**[**lower emissions of carbon and other types of pollution**](https://www.nrdc.org/issues/increase-renewable-energy)**. Wind power generation** capacity in **India** has significantly increased in recent years. As of 29 February 2020 the total installed **wind power** capacity was 37.669 GW, the fourth largest installed **wind power** capacity in the world. **Wind power** capacity is mainly spread across the Southern, Western, and Northern regions.

However, levels of production of wind energy are hard to predict as they rely on potentially unstable weather conditions present at the wind farm. In particular, wind speed is crucial for energy production based on wind, and it may vary drastically over time. Energy suppliers are interested in accurate predictions, as they can avoid overproduction by coordinating the collaborative production of traditional power plants and weather-dependent energy sources.

Here our app can be very useful and helpful. It can not only predict energy but also wind speed and its direction.

**CONCLUSION**

This project help uses in understanding IBM Service in details, from developing to the deployment of the model we learned to use IBM Watson studio and its functions. Node-Red helps us in building our app very easily. Through this project, we also learned to use different IoT Devices and Simulators.

**BIBLIOGRAPHY**

**Team Name:-** Team V.A.P.H.

**Members**:- Vinit Kumar(IIT BHU), Abhinav Kumar Singh(BIT Sindri), Prithviraj Samanta(IIIT Ranchi), Hrishav Kumar Jha (Heritage Institute of Technology)

**Work Title**:- Windbot

**References**:-

* Data set used for training:- <https://www.kaggle.com/berkerisen/wind-turbine-scada-dataset>
* GitHub Repository:- <https://github.com/SmartPracticeschool/SBSPS-Challenge-1678-WindBOT>
* App Link:- <https://node-red-qdxqz.mybluemix.net/ui/#!/0?socketid=Qh0yAMDMHQHoC_EjAABQ>
* Video Explanation:- <https://drive.google.com/file/d/1RteSeDfWqTvcNk-IeViG_BKggu1xGZIx/view>