IoT Based Weather Monitoring System

A project report submitted in partial fulfilment of the requirements of course Wireless and Mobile Communication.

18BEC0696 Vedika Vinod
18BEC0711 Nikhil Subramanian
18BEC0730 Bojanapureddy Guruvishnu
18BEC0735 Savargaonkar Himanshu Aravind

Under the supervision of Prof. Abhijit Bhowmick, School of Electronics Engineering

VELLORE INSTITUTE OF TECHNOLOGY VELLORE 2021

TABLE OF CONTENTS

Sl. No.	Content	Pg. No
1.	Acknowledgements	3
2.	Abstract	4
3.	Introduction	5
4.	Literature Review	6
5.	System Model	9
6.	Methodology	13
7.	Results and Discussion	15
8.	Conclusion	18
9.	References	19

ACKNOWLEDGEMENT

First and foremost, we would like to express our gratitude to our professor Prof. Abhijith Bhowmick. for the continuous support and guidance for our Wireless Mobile Communications project. His guidance and immense knowledge in this field helped our team to accomplish the project within the stipulated timeline. Besides our professor, we would like to thank and acknowledge Dean Sense and the other faculties of sense for giving us the golden opportunity to undergo this project in the field of control systems. We are certain that the skills we acquired during the course of this project will help us in our future endeavours. We thereby acknowledge all the people who were involved in this project in some way or the other.

ABSTRACT

With The growing concern regarding climate change and unpredictable weather patterns, hazard control and mitigation has become challenging as we are unaware of what kind of weather to prepare for. Weather forecasts are made by collecting data about the current state of the atmosphere and using an understanding of atmospheric processes to predict how the atmosphere will evolve. Temperature, humidity, precipitation, air pressure, wind speed, and wind direction are key observations of the atmosphere that help forecasters predict the weather. In this project, we intend to use some of these key parameters to observe the weather pattern and predict the weather for the next few hours using a machine learning model and sending this message to a user wirelessly.

INTRODUCTION

Weather forecasting has been a standout amongst the most experimentally and technologically troublesome tasks over the world in the most recent century. Environmental change has been looking for a great deal of consideration since a long time because of the sudden changes that happen. With the whole world witnessing climate change due to Global Warming, the demand for weather and climate forecast information in support of critical decision-making has grown rapidly during the last decade and will increase even faster in the coming years. The generation and provision of these services has been revolutionized by supercomputers, satellite and remote sensing technology, smart mobile devices. But we are yet to find a cost-effective method that is easily accessible to anyone and makes our lives easier. Using such a weather forecasting device, a user can plan his day, based on whether it is going to be rainy, sunny, humid, etc. Moreover, this device sends the results wirelessly, so that the user gets to know the weather trends of that location remotely. We have used a machine learning model to achieve this.

LITERATURE REVIEW

1. Weather Forecast Prediction: An Integrated Approach for Analyzing and Measuring Weather Data:

Abstract: This project aims to estimate the weather by utilizing predictive analysis. For this reason, analysis of various data mining procedures is needed before apply. This paper introduces a classifier approach for prediction of weather condition and shows how Naive Bayes and Chi square algorithm can be utilized for classification purpose. It further uses a decision tree as shown in Fig 1.

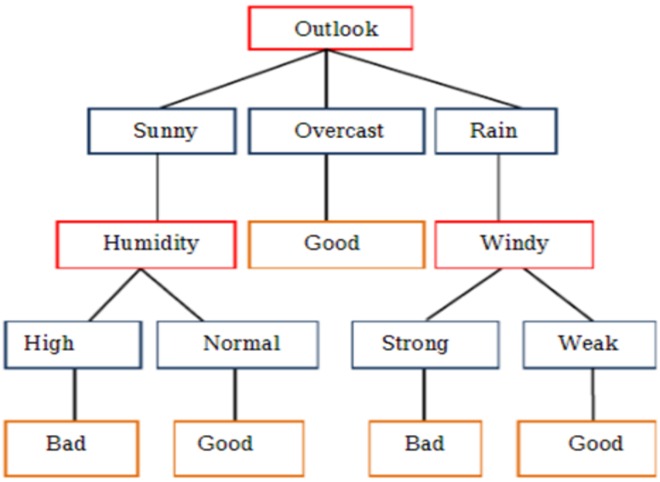


Fig. 1: Decision tree for weather forecast

RESULTS AND CONCLUSION:

This paper works with mix of Naïve Bayes and Chi Square algorithm to predict weather condition. The proposed method produces higher accuracy than the traditional forecasting models.

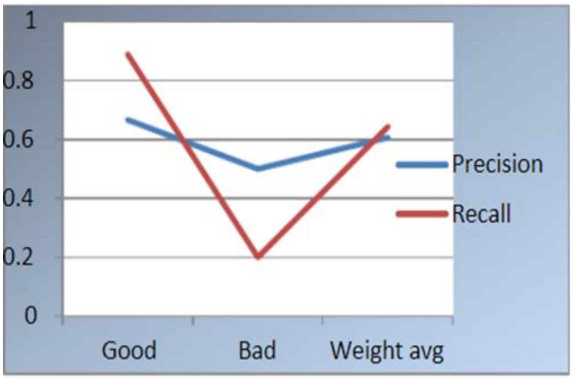


Fig. 2: Performance analysis

2. Artificial Intelligence Revolutionises Weather Forecast, Climate Monitoring and Decadal Prediction

Abstract: The use of AI techniques can lead simultaneously to: (1) a reduction of human development effort, (2) a more efficient use of computing resources and (3) an increased forecast quality. To realise this potential, a new generation of scientists combining atmospheric science domain knowledge and state-of-the-art AI skills needs to be trained. AI should become a cornerstone of future weather and climate observation and modelling systems.

RESULTS AND CONCLUSION:

Traditional NWP models solve an initial value problem, where the knowledge of the current state of the atmosphere—based on observations—is combined with physics-based prognostic equations, which are solved iteratively forward in time. In the new DL approach, the traditional initial value problem can be replaced by an end-to-end trainable DL model, where the forecast step is included in the model optimisation. Such a DL model has the potential to be not only more efficient in terms of human development effort and computing power, but also more performant in terms of forecast quality.

3. Association between weather data and COVID-19 pandemic predicting mortality rate: Machine learning approaches

Abstract: In this work, various regressor machine learning models are proposed to extract the relationship between different factors and the spreading rate of COVID-19. The machine learning algorithms employed in this work estimate the impact of weather variables such as temperature and humidity on the transmission of COVID-19 by extracting the relationship between the number of confirmed cases and the weather variables on certain regions. To validate the proposed method, the researchers collected the required datasets related to weather and census features and necessary prepossessing is carried out.

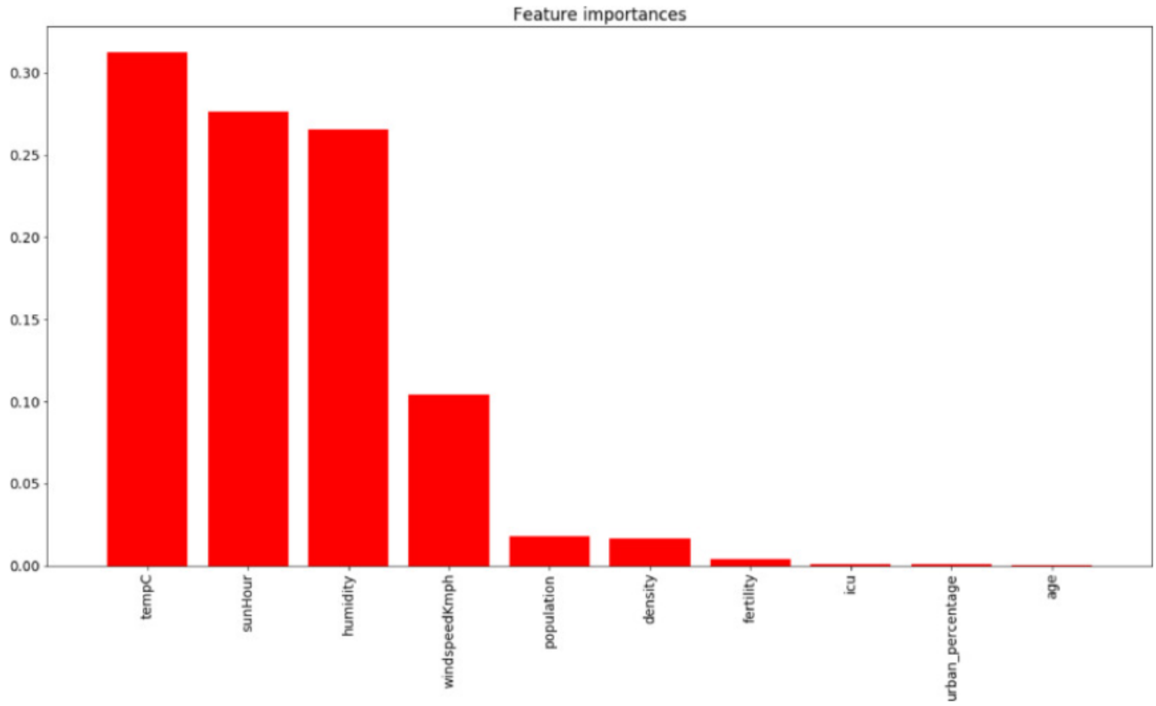


Fig. 3: Features importance for Deaths of Covid-19 cases (17/05/2020).

RESULTS AND CONCLUSIONS:

From the experimental results as shown in fig, it is experimentally proved that the weather variables are more important when compared to other factors such as census feature including population, age, and urban percentage. Thus, from the experimental result, we can conclude that temperature and humidity are important features for predicting COVID-19 death rates, and it does not seem that the ICU beds per 1000 people are an important feature.

SYSTEM MODEL

Our model can be implemented on hardware as shown in the circuit diagram. The connections are made as shown in the circuit diagram. Our hardware only consists of a Node MCU or any other development board and the DHT11 sensor for sending real-time weather parameters

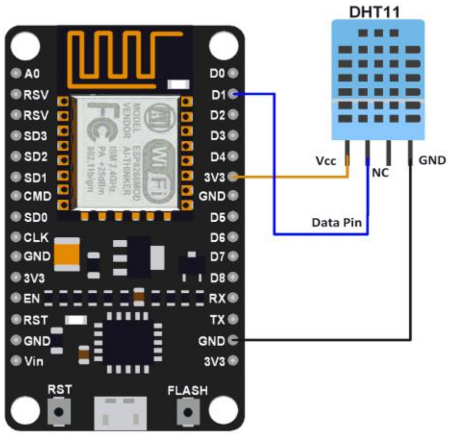


Fig. 4: Circuit diagram

Hardware components used:

1. **NodeMCU:**

   NodeMCU is a low-cost open source IoT platform.[4][5] It initially included firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which was based on the ESP-12 module.

   There are two available versions of NodeMCU as version 0.9 & 1.0 where the version 0.9 contains **ESP-12** and version 1.0 contains **ESP-12E** where E stands for "Enhanced". For this project we have made use of the ESP-12E.

2. **DHT11 sensor:**

   The **DHT11** is a commonly used **Temperature and humidity sensor that** comes with a dedicated NTC to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data.

Technology stack used:

Python, Collaboratory,

Blynk App

Python Libraries used are as shown in Table 1.

Table 1

Sl. no.	**Library used**	**Function**
1.	numpy	For working with arrays. NumPy arrays are stored at one continuous place in memory unlike lists, so processes can access and manipulate them very efficiently. This behavior is called locality of reference in computer science. This is the main reason why NumPy is faster than lists. Also it is optimized to work with latest CPU architectures
2.	keras	Provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library. It is a high-level neural networks library that is running on the top of TensorFlow, CNTK, and Theano. Using Keras in deep learning allows for easy and fast prototyping as well as running seamlessly on CPU and GPU. The following are the advantages of using the keras library: ● **User-Friendly:** Keras has a simple, consistent interface optimized for common use cases which provides clear and actionable feedback for user errors. ● **Modular and Composable:** Keras models are made by connecting configurable building blocks together,

		with few restrictions. ● **Easy To Extend:** With the help of Keras, you can easily write custom building blocks for new ideas and researches. ● **Easy To Use**: Keras offers consistent & simple APIs which helps in minimizing the number of user actions required for common use cases, also it provides clear and actionable feedback upon user error.

Machine learning model:

This project has implemented weather prediction using neural networks. Neural networks is one of the most powerful and widely used algorithms when it comes to the subfield of machine learning called deep learning. A neural network is defined as a computing system that consist of a number of simple but highly interconnected elements or nodes, called 'neurons', which are organized in layers which process information using dynamic state responses to external inputs.

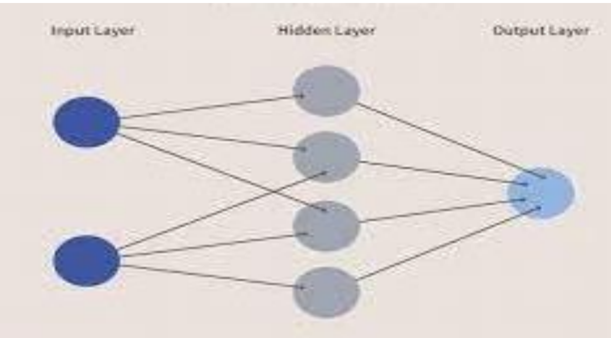


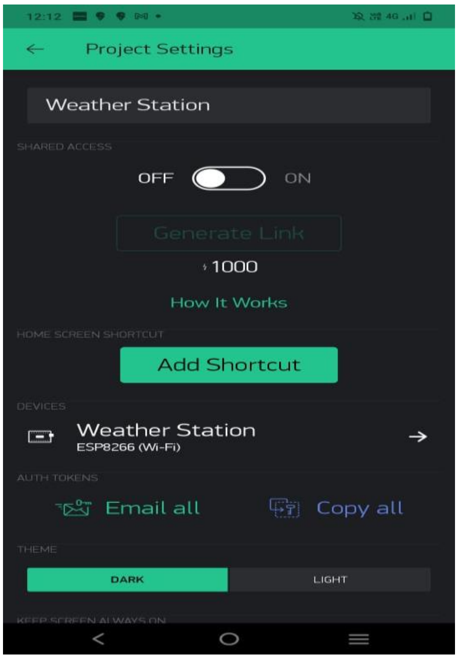
Fig 5: Simple neural network

As our system can measure only 3 parameters, it is very hard to design a AI model around such small number of parameters. Hence as a Proof of Concept, we will be using a dataset with 12 parameters like Temperature, humidity, pressure, relative pressure, location, dew point, rain, altitude, wind speed, etc. Our system can be expanded to 12 parameters also but some data measurements require expensive equipment hence we settled with a dataset.

METHODOLOGY

We made the hardware circuit with the NodeMCU and DHT11 sensor as shown in the system models section. Connect VCC pin on the sensor to the 3.3V pin on the NodeMCU and ground to ground. Also connect Data pin on the sensor to D8 pin of the ESP8266 NodeMCU. Finally, we need to place a pull-up resistor of 10KΩ between VCC and data line to keep it HIGH for proper communication between sensor and NodeMCU.

1. Along with the DHT11 library, we also installed the Blynk library. Blynk is a Platform with **IOS and Android apps to control Arduino, Raspberry Pi and the likes over the Internet**. It's a digital dashboard where you can build a graphic interface for your project by simply dragging and dropping widgets. In the blynk app, we used our login credentials and created a new dashboard called "Weather station".



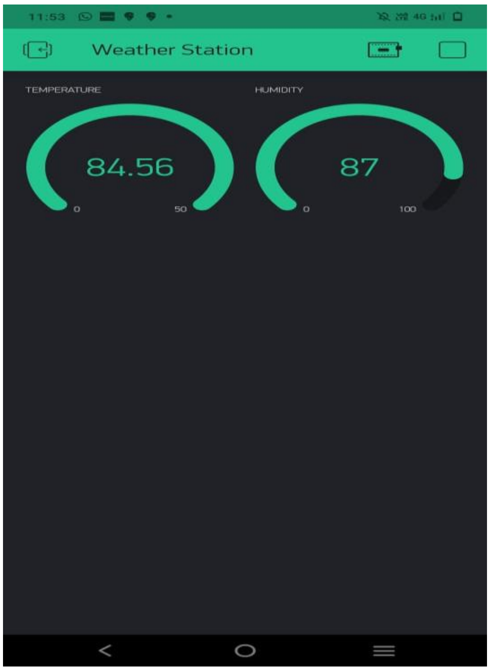


Fig 6: (a) blynk app settings (b)App showing temperature and pressure

Once this dashboard is created, an authorization code is generated, this code is used as the SSID in the arduino code. Further, inside the dashboards we created two widgets namely "temperature", which displays the temperature in Farenheit and "Humidity" which gives the Humidity in percentage.

2. Weather prediction using machine learning:

   The given values are then passed through a machine learning model

   We have used Neural networks for this project. The model we have chosen in the sequential model.

3. The model is trained using the data and then saved into a h5 file. Another program will use the sequential model from this h5 file, make predictions for the weather. Once this is done, the data is given the correct label and saved to the main dataset and made to train again. This label requires human intervention as only a human can verify if the prediction is correct or wrong and add the data label.
4. The Training and predicting as disjointed because training the model requires 3-4 minutes whereas predicting is done in under a minute and running both everytime is waste of time and increases redundancy. By making them disjointed, training can happen in its own pace and prediction separately.

5. Our predicted data is then sent to the ThingsSpeak platform, where we have created a channel called “Weather station” using the corresponding write API key. That will be showing the percentage of thunderstorm, rain, fog and sunshine.

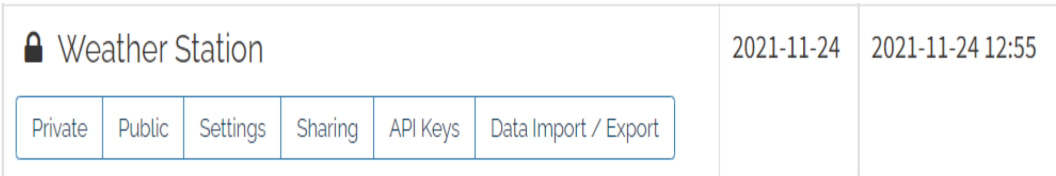
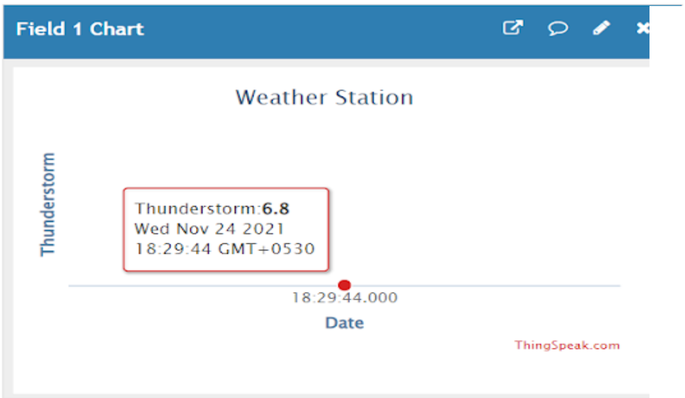


Fig 7: Weather station channel created on Thingspeak

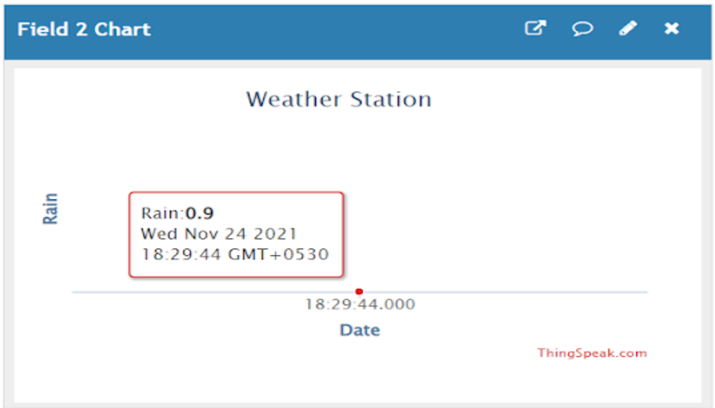
RESULTS AND DISCUSSION

Thingspeak results:

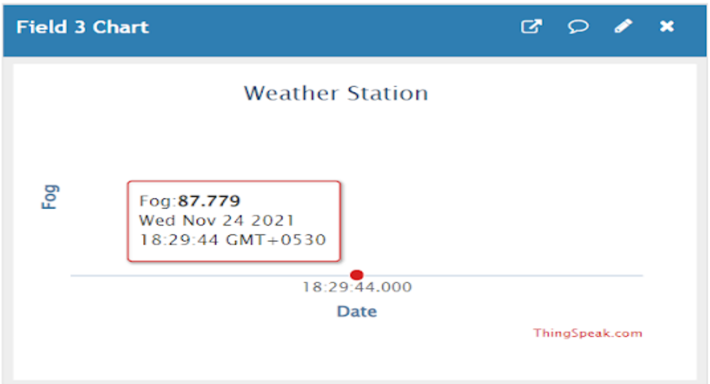
Every time data is pushed in to the cloud, the results appear on the weather station dashboard. Four charts are provided to monitor Thunderstorm, rain, fog and Sunshine.



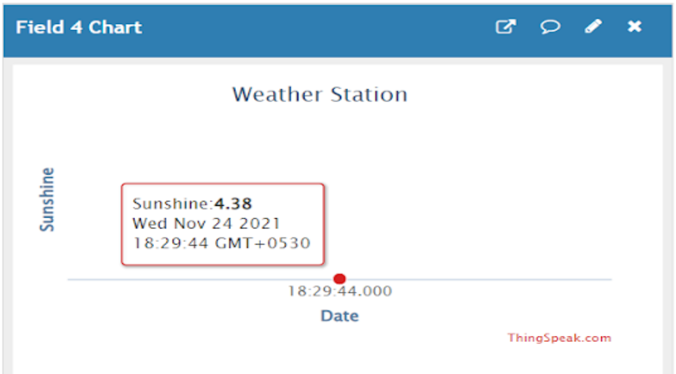
(a)



(b)



(c)



(d)

Fig 8: Thingspeak dashboard

AI model prediction results:

Model Summary:

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_5 (Dense)             (None, 100)               1300

 dense_6 (Dense)             (None, 80)                8080

 dense_7 (Dense)             (None, 60)                4860

 dense_8 (Dense)             (None, 60)                3660

 dense_9 (Dense)             (None, 4)                 244

 activation_1 (Activation)   (None, 4)                 0

=================================================================
Total params: 18,144
Trainable params: 18,144
Non-trainable params: 0
_________________________________________________________________
```

Fig. 9: Model summary

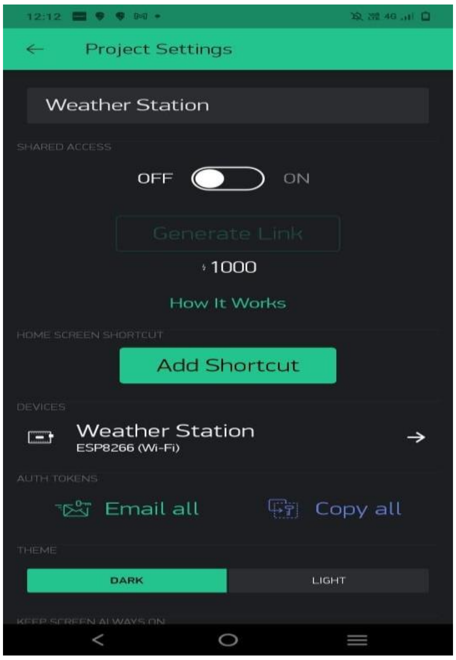
First line is input data, and second line is the prediction in terms of percentage

```
[[   0   22   10   13    9  101   47 1021 1017    3    1   15]]
 [[8.0510724e-04 1.1436009e-03 9.0868831e-01 8.9362919e-02]]
```

Fig. 10: Model predictions

Blynk update:

The blynk dashboard gives the real time temperature in Farenheit and the humidity in percentage. These values are displayed on two separate widgets as shown in figure.



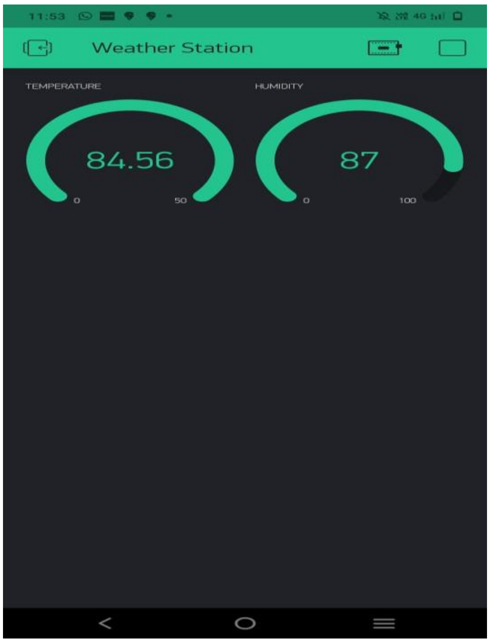


Fig. 11: (a) Blynk settings (b) Blynk dashboard

CONCLUSION

It can be concluded that a neural network is an efficient way of predicting weather. By interfacing the NodeMCU with the Thinkspeak platform and the Blynk app, and further applying the ML model on the collected data, we have been able to calculate the probability in percentage of rain, thunderstorm, fog and storm. With the current scenario of rapid climate change, accurate models to predict the forthcoming weather has become the need of the hour. Adding more features to the current model, aside from temperature and humidity, henceforth adding more neurons to the network will help us predict the weather even more accurately and this model can be used for a variety of applications in various industries like agriculture, manufacturing, healthcare, tourism, etc.

REFERENCES

[1] Munmun Biswas, Tanni Dhoom, Sayantanu Barua (2018) “Weather Forecast Prediction: An Integrated Approach for Analyzing and Measuring Weather Data”

[2] Zohair Malki, El-Sayed Atlam, Aboul Ella Hassanien, Guesh Dagnew, Mostafa A Elhosseini, Ibrahim Gad,(2020) “Association between weather data and COVID-19 pandemic predicting mortality rate: Machine learning approaches”

[3] Steve Dewitte, Jan P. Cornelis, Richard Müller, Adrian Mudreanu,(2021) “Artificial Intelligence Revolutionises Weather Forecast, Climate Monitoring and Decadal Prediction”