Internet of Everything (ITL 702)

RAIN MONITORING BASED ON IOE

B. E. Information Technology

By

Christina Manakkal 40
Aditi Pednekar 41
Anudnya Patil 42
Sherly Mathais 43

Mentor:

Ms. Garima Singh

Assistant Professor

Department of Information Technology
St. Francis Institute of Technology
(Engineering College)

University of Mumbai
2022-23

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources.

We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in this submission.

We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

1. ----------
(Signature)

(Christina Manakkal 40)

2. ----------
(Signature)

(Aditi Pednekar 41)

3. ----------
(Signature)

(Anudnya Patil 42)

4. ----------
(Signature)

(Sherly Mathias 43)

CERTIFICATE

This Internet of Everything Lab Mini-project “Rain Monitoring Based on IOE” by Christina Mannakal, Aditi Pednekar, Anudnya Patil and Sherly Mathias is complete in all respects and was successfully demonstrated on {11th November 2022}.

Name : ----------

Signature :----------

(Internal examiner)

Name : ----------

Signature :----------

(External examiner)

Name : ----------

Signature :----------

(Head of the Department)

Date:

Place:

CONTENTS

Chapter 1
Introduction

Water is an essential resource. Several studies have been conducted to understand rain. The passage of rainfall through a particular area is a dynamic process, it will change the intensity and shape of rain. The measurement and sensing of the amount and type of rain helps us develop a detailed description of rain physically and dynamically. So a better study of our environment and preparing for possible environmental disasters such as droughts or floods. The main aim of water management from rainfall is to reduce economic impact and risk of lives. Most of the developing countries depend on rainfall for their water needs. However if the management of rainfall water is poor, it is difficult to determine water losses to ground ,water recharge and how much is still available etc. Therefore rainfall measurement and monitoring is an important factor. A rainfall monitoring system will continuously measure rainfall and transmit the data to a platform where emergency services or weather experts can alert the residents of the locality, which is threatened by the impending rain related disaster.
The primary objective of the project is to alert its occupants if the rainfall exceeds to a dangerous scale. Reliable rainfall monitoring system is a key for any water system analysis and for operational water system control. Rainfall is measured using gauges which are manually read or automatically send data to a network using a wireless communication system. The system consists of a Tipping Bucket Rain Gauge, an arduino board, Wi-Fi module and temperature as well as humidity sensors. The key hardware here is the Tipping Bucket Rain Gauge which collects and measures the rainfall. The acquired data is processed and transmitted with help of an arduino module where the intended recipient can view it. Tipping bucket rain gauges commonly used for measuring rainfall throughout the world.

Chapter 2
Literature Review

The authors in [1], have analyzed how high rainfall can cause floods in some communities because we need a tool and system which is designed by using Arduino Uno Microcontroller which is integrated with Rain sensor, Humidity, and Temperature sensor to monitor the weather. The authors proposed an application designed to provide warnings and reports on the size of the rainfall level that can minimize the potential for flooding from the beginning.The results of this study indicate that the design of the system with a fuzzy method is very effective, more responsive in providing information about the level of rainy weather.

The authors in [2], have analyzed the cricket match scenario, as it is a very popular game in the country. Due to rain, a major problem that the Cricket stadiums are facing nowadays is that the match gets either the matches are canceled or being delayed and played after the rain. So the authors have proposed a solution by introducing a Rain detector with Alarm and also an Automatic Roofing System for the Cricket Stadiums. This system includes an auto roof which covers the whole stadium. Whenever there is rainfall the rain sensor activates and gives intimation to the Arduino UNO, GSM and it will indicate the LED to glow, buzzer to make sound and the servo motor will automatically close the roof. At the same time a message is sent to the connected mobile device regarding the rain alert. When the rain stops, the roof opens automatically.

The authors in [3], have analyzed that rainwater can be harvested for agricultural purposes and household activities because as water is the main thing in human life we must try to save it and use it for future purposes. The authors proposed a solution in this project that will trigger the alarm when it rains so we can make some actions for rain water harvesting and also to save the rain water for using it later for agriculture in fields.The rain water detector-alarm system will be useful in both domestic and industrial applications. The rain sensor can be made so sensitive that it can detect even the smallest drop of water and triggers the buzzer which is proven to be quite reliable and consistent.

In Table 1 below, we provide the summarized survey results from the research paper studied.

Table 1 Summary of Literature Survey

Sr. no	**Title**	**Methodology**	**Gaps/future scope**
1.	Rain Detection System for Estimate Weather Level using the Mamdani Fuzzy Inference System.	This tool and system is designed by using Arduino Uno Microcontroller which is integrated with Rain sensor, Humidity, and Temperature sensor to monitor the weather. Analog and digital input data from Arduino are sent one by one to the	Implementation of weather forecasting requires accurate data from reliable sources to be used as material forecasting in the future. The use of forecasting in

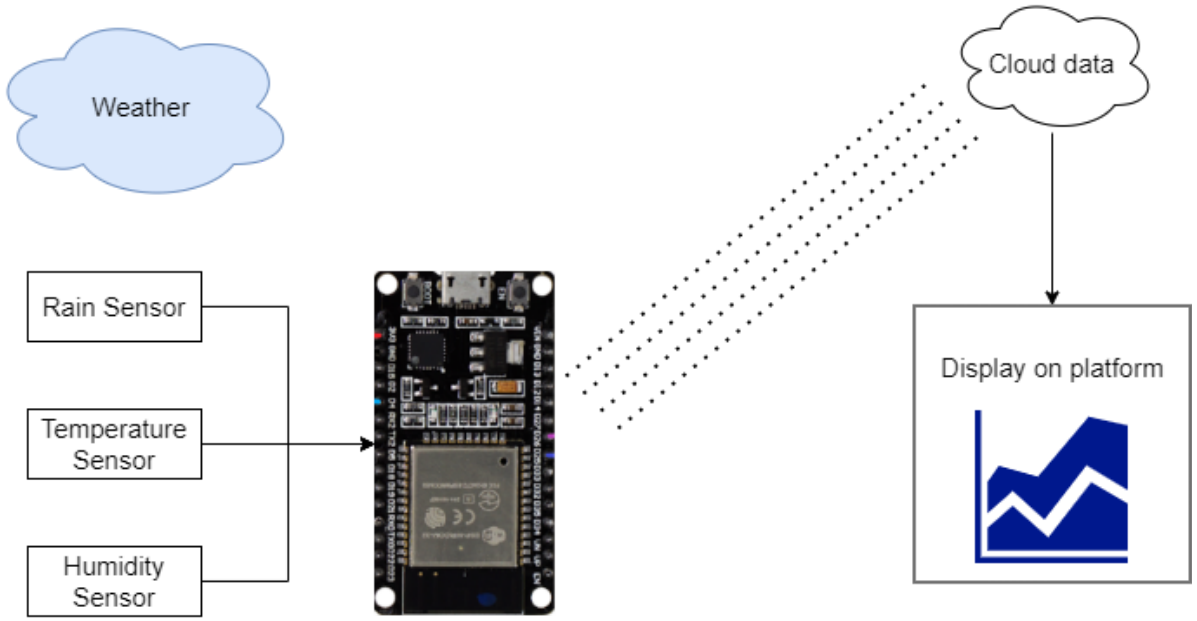
		Fuzzy Interface using serial protocols, the data processed using the Fuzzy Algorithm is used comparing the results of the Rain sensor, Humidity and Temperature sensor outputs.	the rainfall detection system may reduce the accuracy of the initial concept by using the Fuzzy Mamdani method to determine a precise and real time rainfall decision level.
2.	Rain Detector with Alarm	This paper is based on the project which is an embedded system consisting of a Rain drop sensor, Arduino Uno board and a SIM900 GSM module. Arduino Uno receives and processes the data from the sensors. The GSM unit acts as an interface between the Arduino and the user's mobile.The rain sensor will be placed at an open place. As soon as it senses the rain it will pass information to the Arduino which will in turn make the LED to glow, Buzzer to make sound and the alert message will be sent in turn the Servo Motor will automatically close the roof.	This Project does not indicate the rate at which rain falls on the roof. Until the detector is wet, a signal is issued or the LED lights up. The detector will not work until it is wet if something happens.
3.	Smart Rain Detector Using Arduino	The elements of this system includes Arduino Uno, Rain Sensor, Sensing Pad, Rain Module and Buzzer. This system works in such a way that, when there is rain, the rainwater acts as a trigger, which switches on the buzzer.	To use more appropriate rain sensor and make precise automatic rain sensing system with the help of a microcontroller based system to implement some security features for farmers

Chapter 3
Problem Statement

To build a rain sensing/monitoring smart device which will sense the environment parameters like humidity, temperature and provide probability of rain by conducting analysis..

Chapter 4
System Design and Requirements

System Design



System requirements

Hardware Requirements:

1. ESP32 Module (NodeMCU)
2. FC-37 Rain sensor
3. DHT11 Humidity & Temperature Sensor
4. Jumper Wires
5. Breadboard

Software Requirements:

1. OS: Windows 7 or above
2. Browser: Chrome/Firefox/Edge/Brave
3. Application: Arduino IDE
4. MathWorks ThingSpeak

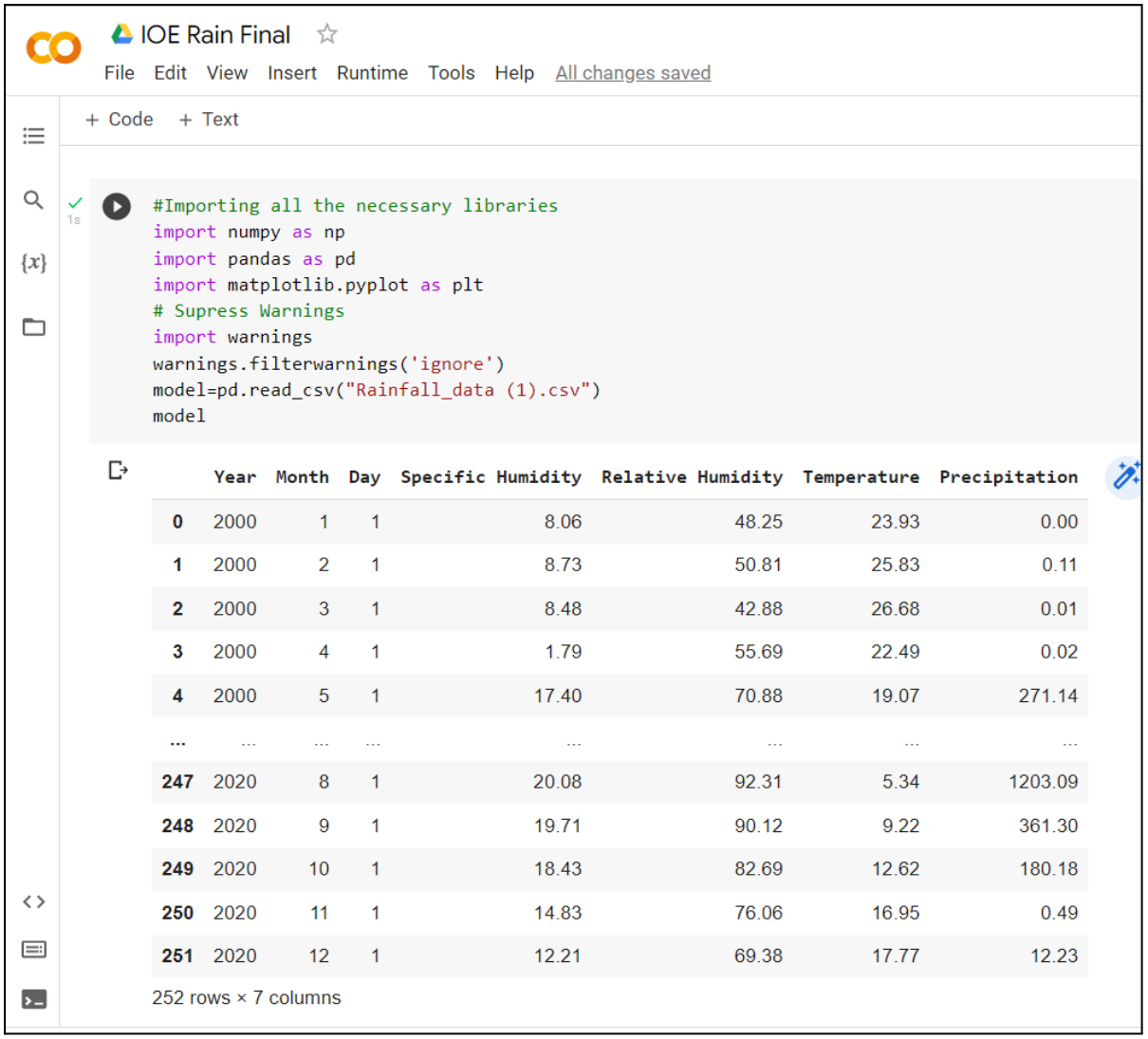
Chapter 5
Data Analytics

Regression analysis is a set of statistical methods used for the estimation of relationships between a dependent variable and one or more independent variables. It can be utilized to assess the strength of the relationship between variables and for modeling the future relationship between them.

In this project, linear regression is used and the dataset is divided in the ratio of 70:30.

The Dependent variable is precipitation and the independent variables are year,month,day,specific humidity,relative humidity and temperature. We can calculate probability of rain through precipitation.

Performance parameters are rmse and mean absolute error percentage.



	Year	Month	Day	Specific Humidity	Relative Humidity	Temperature	Precipitation
0	2000	1	1	8.06	48.25	23.93	0.00
1	2000	2	1	8.73	50.81	25.83	0.11
2	2000	3	1	8.48	42.88	26.68	0.01
3	2000	4	1	1.79	55.69	22.49	0.02
4	2000	5	1	17.40	70.88	19.07	271.14
...	...	...	...	...	...	...	...
247	2020	8	1	20.08	92.31	5.34	1203.09
248	2020	9	1	19.71	90.12	9.22	361.30
249	2020	10	1	18.43	82.69	12.62	180.18
250	2020	11	1	14.83	76.06	16.95	0.49
251	2020	12	1	12.21	69.38	17.77	12.23

252 rows × 7 columns

```
model.head(5)
```

	Year	Month	Day	Specific Humidity	Relative Humidity	Temperature	Precipitation
0	2000	1	1	8.06	48.25	23.93	0.00
1	2000	2	1	8.73	50.81	25.83	0.11
2	2000	3	1	8.48	42.88	26.68	0.01
3	2000	4	1	1.79	55.69	22.49	0.02
4	2000	5	1	17.40	70.88	19.07	271.14

```
[93] model.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 252 entries, 0 to 251
Data columns (total 7 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Year                252 non-null    int64  
 1   Month               252 non-null    int64  
 2   Day                 252 non-null    int64  
 3   Specific Humidity   252 non-null    float64
 4   Relative Humidity   252 non-null    float64
 5   Temperature         252 non-null    float64
 6   Precipitation       252 non-null    float64
dtypes: float64(4), int64(3)
memory usage: 13.9 KB
```

```
model.describe()
```

	Year	Month	Day	Specific Humidity	Relative Humidity	Temperature	Precipitation
count	252.000000	252.000000	252.0	2.520000e+02	2.520000e+02	2.520000e+02	252.000000
mean	2010.000000	6.500000	1.0	6.145007e+14	2.116867e+12	8.130833e+08	206.798929
std	6.067351	3.458922	0.0	9.636212e+15	2.843746e+13	1.290730e+10	318.093091
min	2000.000000	1.000000	1.0	8.600000e-01	1.940000e+00	4.730000e+00	0.000000
25%	2005.000000	3.750000	1.0	9.685000e+00	5.119000e+01	1.063000e+01	0.402500
50%	2010.000000	6.500000	1.0	1.458500e+01	6.659000e+01	1.687500e+01	11.495000
75%	2015.000000	9.250000	1.0	1.887500e+01	8.551750e+01	2.220250e+01	353.200000
max	2020.000000	12.000000	1.0	1.529660e+17	4.425470e+14	2.048970e+11	1307.430000

```
[95] model.shape
```

```
(252, 7)
```

```
[96] model.columns
```

```
Index(['Year', 'Month', 'Day', 'Specific Humidity', 'Relative Humidity',
       'Temperature', 'Precipitation'],
      dtype='object')
```

```
from sklearn.preprocessing import MinMaxScaler
# Creating an instance of the sklearn.preprocessing.MinMaxScaler()
scaler = MinMaxScaler()
# Scaling the Price column of the created dataFrame and storing
# the result in Scaled Column
model[["ScaledPreci"]] = scaler.fit_transform(model[["Precipitation"]])
print(model)
```

```
      Year  Month  Day  Specific Humidity  Relative Humidity  Temperature  \
0     2000      1    1               8.06              48.25        23.93
1     2000      2    1               8.73              50.81        25.83
2     2000      3    1               8.48              42.88        26.68
3     2000      4    1               1.79              55.69        22.49
4     2000      5    1              17.40              70.88        19.07
..     ...    ...  ...                ...                ...          ...
247   2020      8    1              20.08              92.31         5.34
248   2020      9    1              19.71              90.12         9.22
249   2020     10    1              18.43              82.69        12.62
250   2020     11    1              14.83              76.06        16.95
251   2020     12    1              12.21              69.38        17.77

      Precipitation  ScaledPreci
0              0.00     0.000000
1              0.11     0.000084
2              0.01     0.000008
3              0.02     0.000015
4            271.14     0.207384
..              ...          ...
247         1203.09     0.920195
248          361.30     0.276344
249          180.18     0.137812
250            0.49     0.000375
251           12.23     0.009354

[252 rows x 8 columns]
```

```
[98] #define X=features and Y=target variables
     Y=model['ScaledPreci']
     # X=model[['Year', 'Month', 'Day', 'Specific Humidity', 'Relative Humidity','Temperature']]
     X=model[['Year', 'Month','Day','Relative Humidity','Temperature']]
```

```
[109] Y.shape, X.shape
```

```
((252,), (252, 5))
```

```
[101] from sklearn.model_selection import train_test_split
      X_train,X_test,y_train,y_test=train_test_split(X,Y,train_size=0.7,random_state=5)
```

```
X_train.shape,X_test.shape,Y_train.shape,Y_test.shape
```

```
((176, 5), (76, 5), (176,), (76,))
```

```
import statsmodels.api as sm
# Add a constant to get an intercept
X_train_sm = sm.add_constant(X_train)
# Fit the resgression line using 'OLS'
lr = sm.OLS(y_train, X_train_sm).fit()
# Print the parameters,i.e. intercept and slope of the regression line obtained
lr.params
```

```
Year                 3.888127e-04
Month               -1.827015e-02
Day                 -7.889811e-07
Relative Humidity   -1.835272e-15
Temperature         -3.143700e-02
dtype: float64
```

```
#Performing a summary operation lists out all different parameters of the regression line fitted
print(lr.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:            ScaledPreci   R-squared:                       0.620
Model:                            OLS   Adj. R-squared:                  0.613
Method:                 Least Squares   F-statistic:                     93.42
Date:                Tue, 01 Nov 2022   Prob (F-statistic):           6.47e-36
Time:                        06:10:13   Log-Likelihood:                 89.963
No. Observations:                 176   AIC:                            -171.9
Df Residuals:                     172   BIC:                            -159.2
Df Model:                           3
Covariance Type:            nonrobust
==========================================================================================
                             coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------
Year                       0.0004   2.45e-05     15.843      0.000       0.000       0.000
Month                     -0.0183      0.004     -4.919      0.000      -0.026      -0.011
Day                   -7.89e-07   2.29e-07     -3.442      0.001   -1.24e-06   -3.37e-07
Relative Humidity -1.835e-15   1.63e-15     -1.124      0.263   -5.06e-15    1.39e-15
Temperature           -0.0314      0.002    -16.344      0.000      -0.035      -0.028
==============================================================================
Omnibus:                       32.069   Durbin-Watson:                   2.134
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               67.824
Skew:                           0.826   Prob(JB):                     1.87e-15
Kurtosis:                       5.553   Cond. No.                     2.28e+15
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.28e+15. This might indicate that there are
strong multicollinearity or other numerical problems.
```

```
[105] # Add a constant to X_test
      X_test_sm = sm.add_constant(X_test)
      # Predict the y values corresponding to X_test_sm
      y_pred = lr.predict(X_test_sm)
      y_pred.head()
```

```
      247    0.471366
      168    0.056836
      127    0.444529
      83     0.008367
      216    0.103347
      dtype: float64
```

```
[106] #Imporitng libraries
      from sklearn.metrics import mean_squared_error
      from sklearn.metrics import r2_score
```

```
      from sklearn.metrics import mean_absolute_error,mean_absolute_percentage_error
      mean_absolute_error(y_test,y_predict)
      mean_absolute_percentage_error(y_test,y_predict)
```

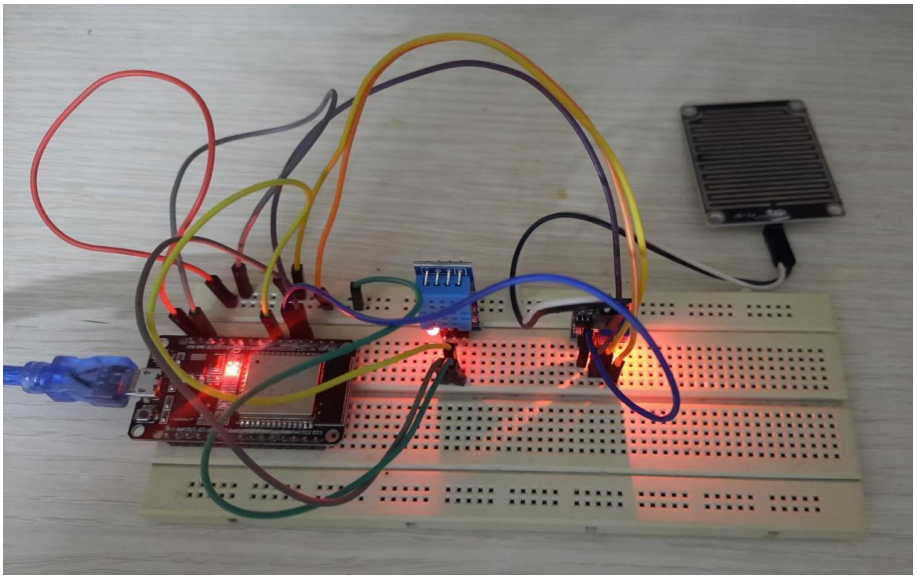
```
      3.8221283334379476e+23
```

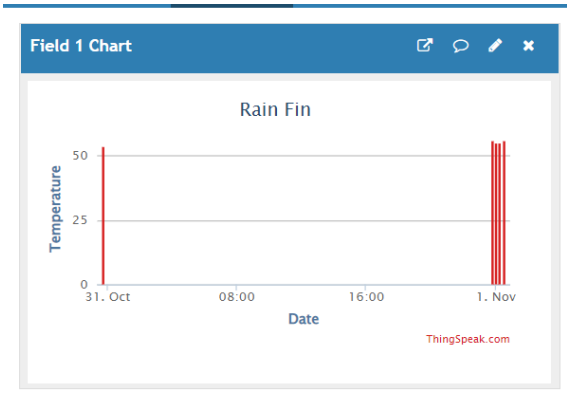
```
      pickle.dump(Lr, open('model.pkl','wb'))
      model = pickle.load(open('model.pkl','rb'))
      #Y,M,D,H,T
      pop = model.predict([[2020,1,1,48.25,23.93]])
      pop
```

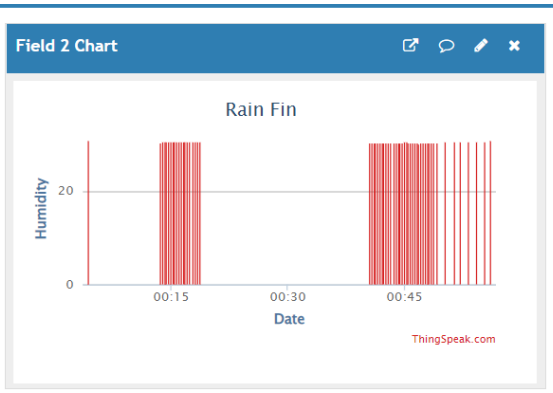
```
      array([0.04235236])
```

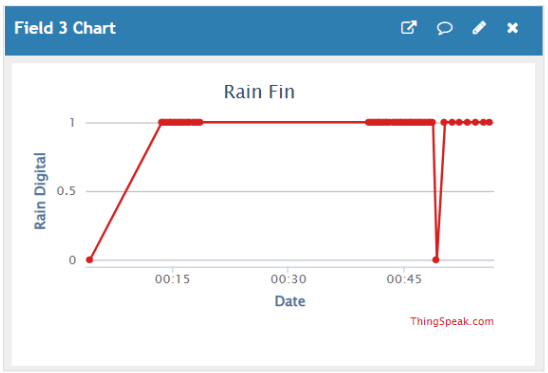
Chapter 6
Results

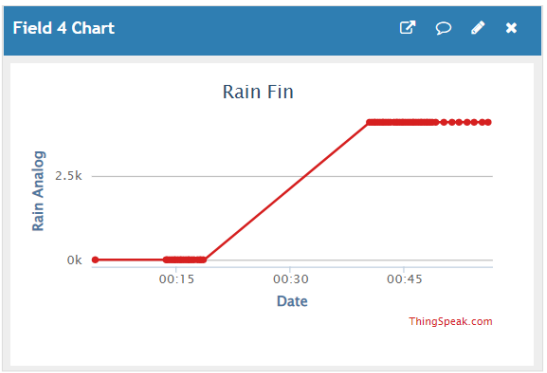
Setup:

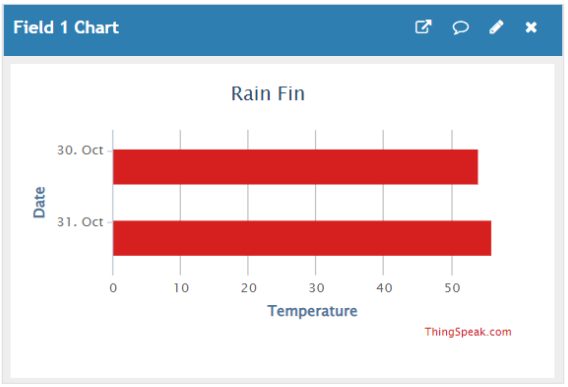


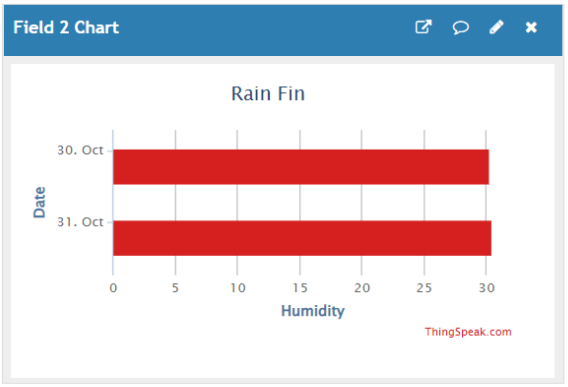


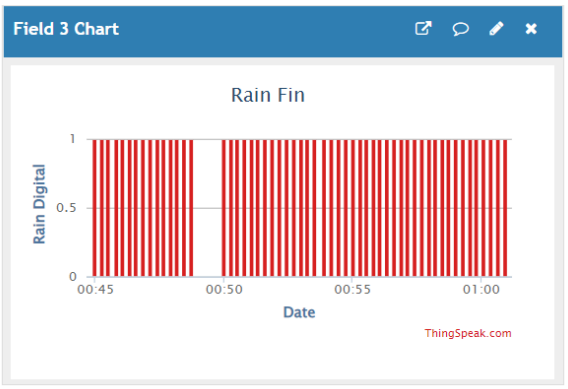


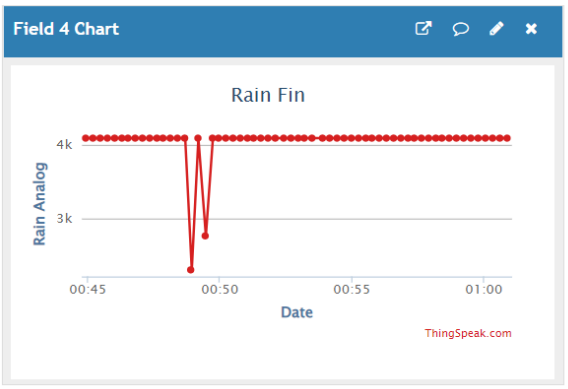




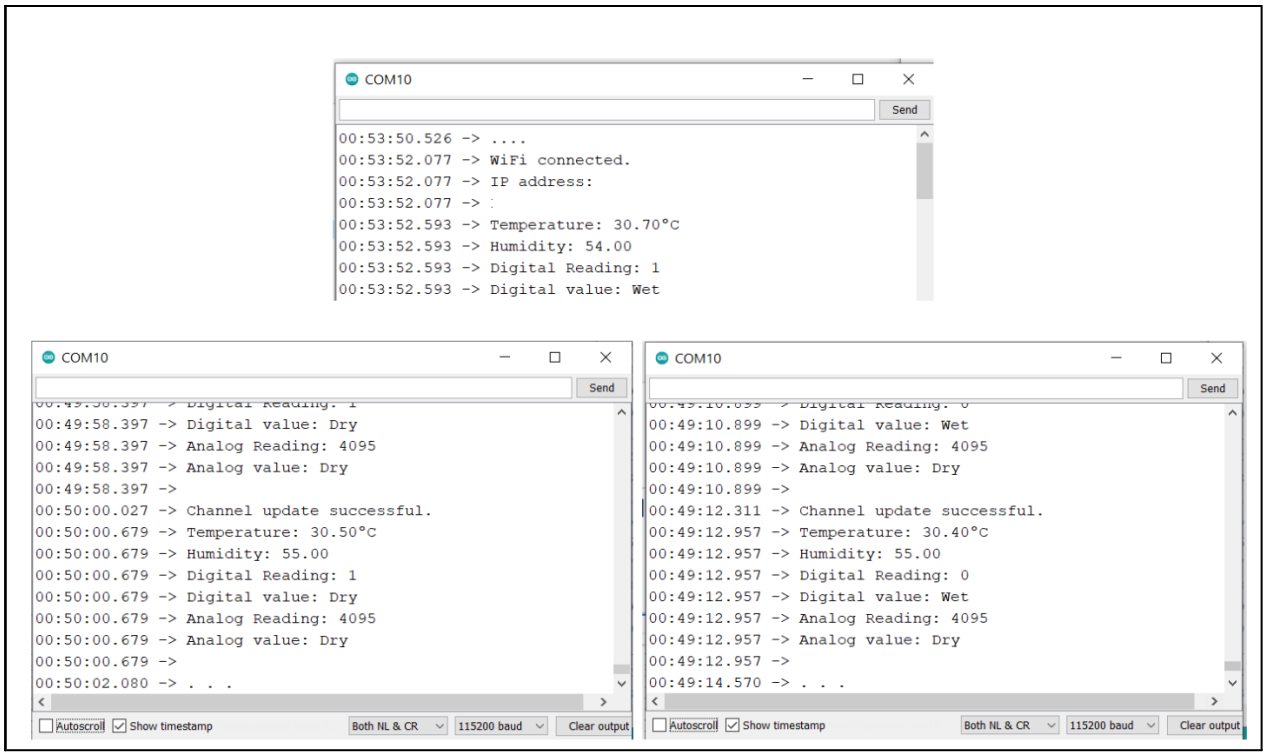








Output reading on Arduino IDE Serial Monitor with baudrate 115200:



Chapter 7
Conclusion and Future Scope

In this project we built a Rain Sensing/Monitoring smart device which senses the environment and alerts the user about rain and conducts analysis. This project introduces an intelligent rain sensing/ monitoring system in the IoT platform which would help users by alerting them about rain so they can take appropriate actions. This rain monitoring system will display the data collected from the sensor using thingspeak as an analysis platform. The result is displayed in the form of a graph.

This system can be improved by adding additional mechanisms like a shutter that automatically closes when the sensor detects rain. We can add LEDs that can indicate the probability of rain i.e green LED will glow if it is not going to rain anytime soon and red LED will glow if there is high probability of rain.

References

1. A. Y. Ardiansyah, R. Sarno, and O. Giandi, "Rain detection system for estimate weather level using Mamdani Fuzzy Inference System," *2018 International Conference on Information and Communications Technology (ICOIACT)*, 2018.

2. V. Divakar, Chithritha G R, Darshan V, "Rain Detector with Alarm," *International Advanced Research Journal in Science, Engineering and Technology*, vol. 8, no. 7, pp. 86–89, Jul. 2021.

3. A. Syed, "Smart Rain Detector Using Arduino." *SSRN Electronic Journal*, pp. 1-8, 2021, doi: 10.2139/ssrn.3918326.

4. A. F. Pauzi and M. Z. Hasan, "Development of IoT Based Weather Reporting System." *IOP Conference Series: Materials Science and Engineering*, vol. 917, no. 1, pp. 012032, 2020, doi: 10.1088/1757-899x/917/1/012032.

5. N. K. A. Appiah-Badu, Y. M. Missah, L. K. Amekudzi, N. Ussiph, T. Frimpong, and E. Ahene, "Rainfall Prediction Using Machine Learning Algorithms for the Various Ecological Zones of Ghana." *IEEE Access*, vol. 10, pp. 5069-5082, 2022, doi: 10.1109/access.2021.3139312.

6. N. Akua Opoku. "ML-Rainfall prediction using linear regression." https://www.educative.io.https://www.educative.io/answers/ml-rainfall-prediction-using-linear-regression (accessed: Oct. 31, 2022).