SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

WINTER SEMESTER - 2022

Reverse cloud seeding

A Report

submitted by

NAVDEEP SUREKA

(Reg. No.: 19BCE2679)

CSE3009 – Internet of Things – J Component

B1 Slot

Faculty Name – Dr.A.Saraswathi Priyadharshini,

Department of Analytics,

SCOPE

VIT UNIVERSITY (Estd. u/s 3 of UGC Act 1956) VELLORE ■ CHENNAI www.vit.ac.in

TABLE OF CONTENTS

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Abstract

To develop the reverse cloud seeding and rain monitoring system using IOT control devices to effectively predict the rain using weather prediction and launch the planned payload to control the rainfall in the particular area. The IoT system will allow different rockets to communicate and initiate the launch sequence and also provide real-time location of the rocket after launch for a successful recovery.

Issue Discussed

A lot of farmers depend on rainfall for the crop production and do not use irrigation methods at all. During droughts or summer, the crops get affected. We need to devise a method to predict the rainfall so as to help the farmers to take preventive measures for heavy rains and also increases chances of rainfall in dry regions of the country.

We as a team aimed to develop and test a model rocket made with readily available equipment and work on its efficiency and cost to solve the given statement.

To help the farmers in such a situation, a well known way known as CLOUD SEEDING can be applied. Cloud seeding is a way to artificially tweak rain. It is also known by other terms such as man-made precipitation enhancement, artificial weather modification, rainmaking and so on.

It can be achieved to change the weather conditions to suit the amount of precipitation as per requirement.

Brief Idea

Cloud seeding alters the moisture available in a cloud to create additional rainfall. This is done by dispersing silver iodide or sodium salts into the clouds and affects the physical processes occurring inside it. Silver iodide is affective because it is similar to ice crystals. These substances can be easily delivered into the clouds by our ground-based rocket. These substances attract water vapor to the cloud and make larger water droplets which in turn make the cloud heavy and increase the chance of rain to a very high extent.

The rocket used to implement this, is a completely self-made rocket at a very cheap price and high usage capacity (reuse). It helps in restoring the ground water in dry areas. It also has hail suppression as one of its applications because it increases the number of smaller hailstones which dry change to droplets before hitting the ground.

With our weather model, we can not only make it rain whenever and wherever we want but also be able to predict the rainfall in an area beforehand. The model will be able to collect local data and simultaneously collect national data. Then it will go through regional weather to predict in 2 days advance weather it will rain or not and if it is going to rain, our system will first launch a sensor rocket which will measure the wind speed and direction in the lower atmosphere then it will tell the direction from which the clouds may come after which since all the system has rockets ready at the nearest reservoir or any place where rainfall won't do any harm or is needed desperately, we will be able to launch rockets at specific intervals for specific uses.

Model Rocket

The model rocket is a solid-state or solid propellant rocket which uses 'sugar propellant' which refers to its composition. This fuel is easy to produce and use in a basic rocket system. The propellant includes a mixture of Potassium Nitrate (KNO3) and sugar (for sucrose). These 2 compounds are average chemicals for burning but together make a good propellant mix. Potassium Nitrate and Sucrose were mixed in various proportions but the best and most efficient mix which burnt spontaneously and at a low ignition temperature is in the ratio of 65:35 of KNO3 and sucrose.

The approximate chemical reaction between the 2 chemical compounds is also given below:

$$48\ KNO_3 + 5\ C_{12}H_{22}O_{11} \rightarrow 24\ K_2CO_3 + 24\ N_2 + 55\ H_2O + 36\ CO_2$$

Advantage of using ROCKETS

Cloud seeding can be achieved using either an airplane or rocket, but using rocket is a best option because not only is it way cheaper but also a much efficient and quicker method. Helicopters can't be used as they will disrupt cloud formation which completely fails the purpose of cloud seeding.

The rocket is made with an effective cost of 200–250 rupees only.

Our model works well for Lower Earth Orbit (LEO) because of the solid-state propellant.

Literature Survey

1) COMMUNICATION ESTABLISHMENT

TITLE : PROXIMITY AWARENESS AND FAST CONNECTION ESTABLISHMENT IN BLUETOOTH

A SYMMETRIC TECHNIQUE FOR ESTABLISHING AD HOC CONNECTIVITY IS INTRODUCED WHICH IMPOSES EACH NODE TO ALTERNATE BETWEEN THE "SENDER" AND "RECEIVER" STATE IN A RANDOM FASHION.

ADVANTAGES

Easy to implement as the chip used is very affordable and the reliability is good enough.

DISADVANTAGES

Have to set a time out for all devices as the comunication is not parallel.

2) CLOUD STORAGE

TITLE : RESEARCH ON CLOUD DATA STORAGE TECHNOLOGY AND ITS ARCHITECTURE IMPLEMENTATION

ANALYZES THE CLOUD DATA STORAGE TECHNOLOGY GFS(GOOGLE FILE SYSTEM)/HDFS(HADOOP DISTRIBUTED FILE SYSTEM)

ADVANTAGES

The data processing is fast and we can use google cloud to perform analyzes without additional code .

DISADVANTAGES

Based on google cloud and using the system with other provider is hard.

3) WEATHER PREDICTION

TITLE : IOT BASED AUTOMATED WEATHER REPORT GENERATION AND PREDICTION USING MACHINE LEARNING

THE MODEL WILL SENSE DIFFERENT WEATHER PARAMETERS LIKE

PRESSURE, HUMIDITY, RAINFALL, TEMPERATURE, DUST PARTICLES AND LIGHT ETC. AND WILL DISPLAY CURRENT DAY'S TEMPERATURE ON A WEB

PORTAL. THE MODEL WILL ALSO BE ABLE TO PREDICT THE MAXIMUM AND MINIMUM TEMPERATURES OF THE NEXT DAY USING PAST THREE

ADVANTAGES

It used both ground based data and data from weather povider and based on both data, it gives a good balance of weather.

DISADVANTAGES

But for our use, the system will be very much bias on the online prediction as on-ground sensors

4) GPS TRACKING

TITLE : IMPLEMENTATION OF GPS BASED OBJECT LOCATION AND ROUTE TRACKING ON ANDROID DEVICE

THIS PROJECT DEVELOPS A TRACKING / MONITORING ANDROID APPLICATION (MOBILE) USING OBJECT GPS DEVICES TO ASCERTAIN ITS CURRENT LOCATION, AND PREVIOUS LOCATION AT SPECIFIED INTERVALS, THIS SYSTEM UNLIKE PREVIOUS TRACKING SYSTEM WILL GIVE USER THE ABILITY TO CREATE BOOKMARK OF CURRENT LOCATION AND ABILITY TO ROUTE BACK TO THAT LOCATION FROM ANYWHERE USING GOOGLE MAPS API'S IN CASE THEY CAN'T REMEMBER THE PRICES LOCATION.

ADVANTAGES

The system have an tracking based on android endpoints so we can easily implement it using simple API call.

DISADVANTAGES

Not used for arial object tracking and the hard impact is not yet tested.

5) RAIN PREDICTION

TITLE: RAINFALL PREDICTION USING MACHINE LEARNING AND NEURAL NETWORK

THERE ARE TWO
TECHNIQUES TO PREDICT RAINFALL. THE FIRST ONE IS MACHINE LEARNING APPROACH, WHICH INCLUDES LASSO REGRESSION. THE SECOND ONE IS NEURAL NETWORK APPROACH. THIS SYSTEM FIRST COMPARES BOTH THE PROCESS AND THEN ACCORDINGLY GIVES RESULT WITH THE BEST ALGORITHM.

ADVANTAGES

As it uses Lasso regression , the data produced can be very much compressed and can be easily communicated.

DISADVANTAGES

The implementatin of code is not avaliable and the reader have to code the stuff based on the theory.

Proposed Work

Use a ground-based weather monitoring station and based on data from the ground instrument and weather forecast prediction, our model will give a better rain prediction and using the data we can calculate the payload capacity.

Three- step process

A) Historical data ->Historical data from the weatherstack API using a node-express app

B) Sensor data -> Data from ground-based sensor.

C) Machine learning -> Based on both data, ML algo will make the rain predictions and bootstrap the rockets.

WorkFlow

1) Create the API: Make the weatherstack API as a standalone application

2) Create data collection:System to store and manage the data from the ground-based hardware.

3) Machine learning: Using flask to host pre-trained ML model

4) Configuring rockets: Based on the predictions, data will be sent to rockets to auto-configure

Machine learning to predict weather

** For the current situation, Indian weather observatory conduct a traditional weather forecasting. There are four common methods to predict weather. The first method is climatology method that is reviewing weather statistics gathered over multiple years and calculating the averages.The second method is analog method that is to find a day in the past with weather similar to the current forecast. The third method is persistence and trends method that has no skill to predict the weather because it relies on past trends. The fourth method isnumerical weather prediction the is making weather predictions based on multiple conditions in atmosphere such as temperatures, wind speed, high-and low-pressure systems, rainfall, snowfall and other conditions.So,there are many limitations of these traditional methods. Not only It forecasts the temperature in the current month at most, but also it predicts without using machine learning algorithms.Therefore, my project is to increase the accuracy and predict weather in the future at least one month through applying machine learning techniques.
**

Objective (Brief)

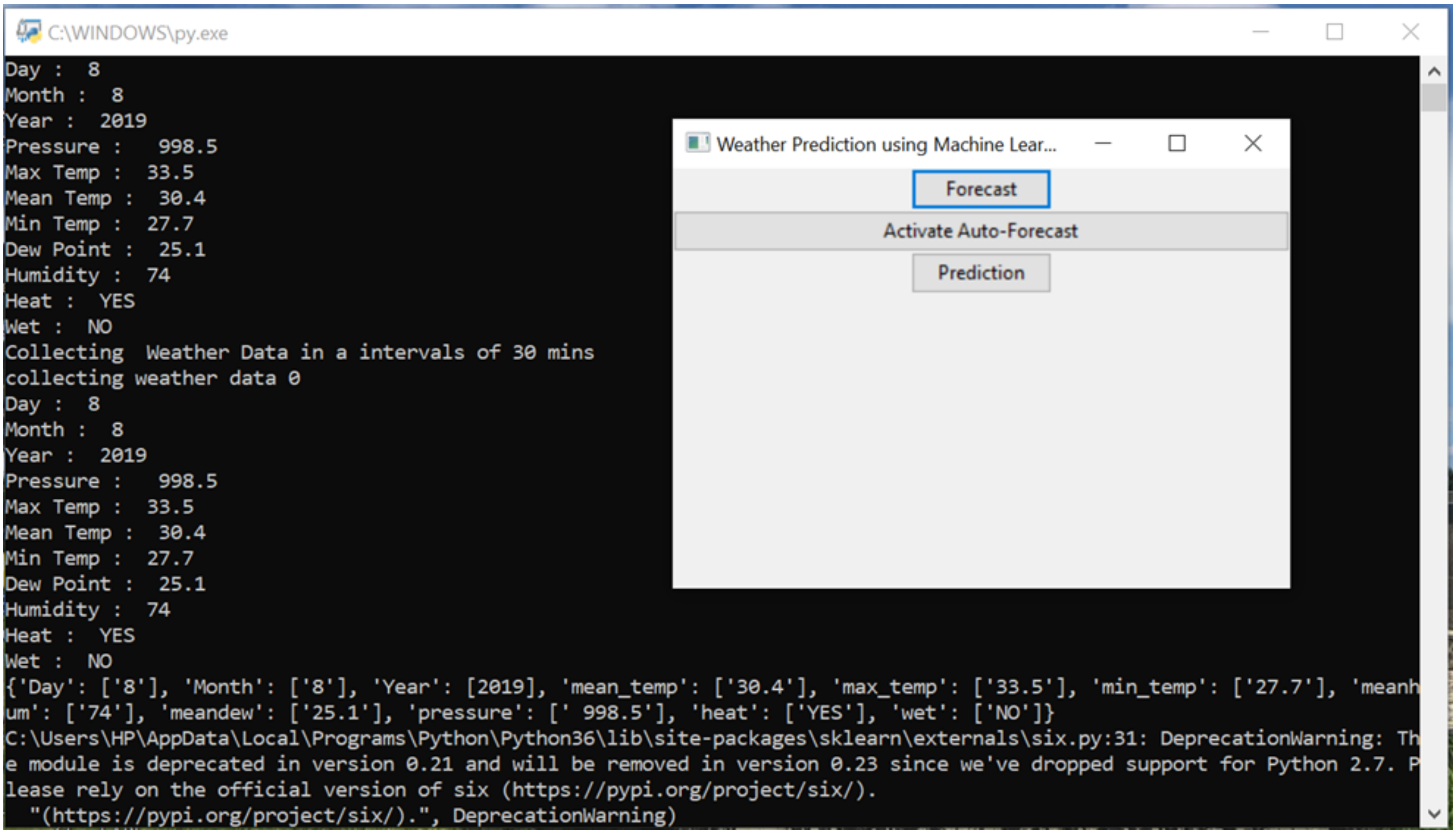
** There are two purposes of my project. One of the purposeis to forecast the status of weather of specific day of a year. I will demonstrate the result through using decision tree regression and show the output for the status of wet or heat. Another aim is to predict the temperature using different algorithms like linear regression, random forest regression and K-nearest neighbor regression. The output value should be numerical based on multiple extra factors like population density and air health quality.
**

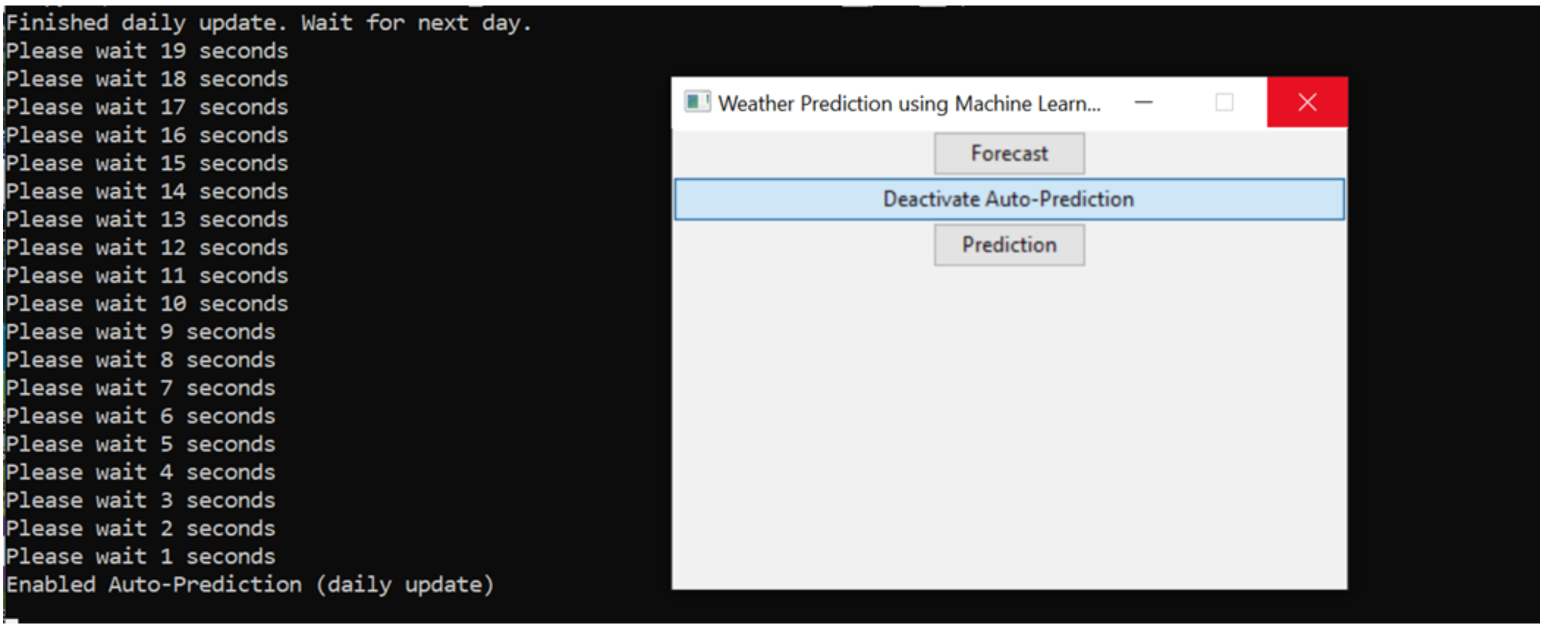
Data Description:

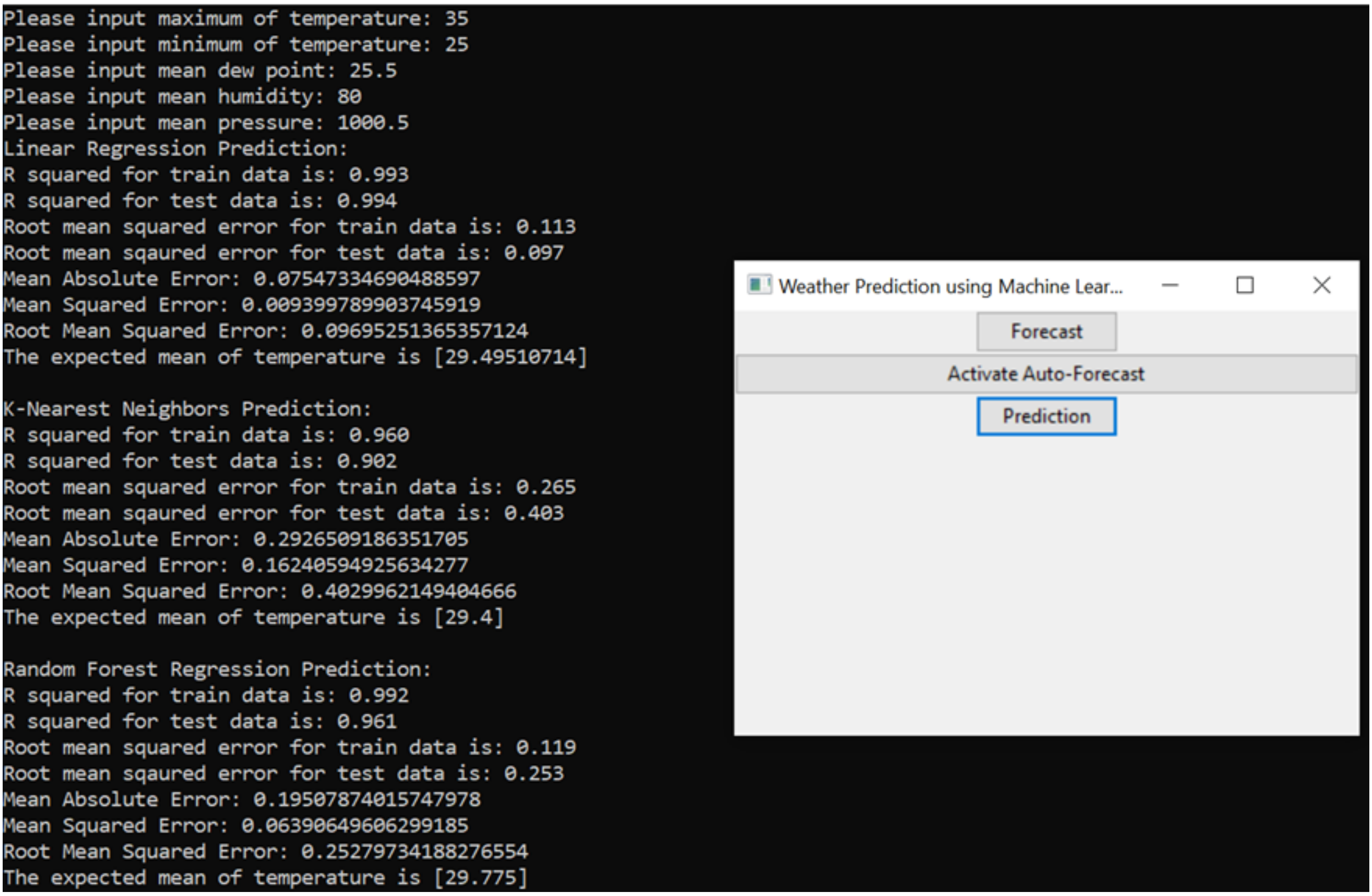
** mean_temp: mean air temperature
max_temp: mean daily maximum air temperature
min_temp: mean daily minimum air temperature
meanhum: mean relative humidity
meandew: mean dew point temperature
pressure: mean daily air pressure
heat: true when mean air temperature is over or equal to 30
wet: true when mean relative humidity is over or equal to 80
Mean_cloud: mean cloud
population: population density
Sunshine_hour: mean number of hour of sunshine
Wind_direction: mean wind direction
Wind_speed: mean wind speed
Air_health_quality: mean daily air health quality
**

Running the model on locally

Outputs and Results:

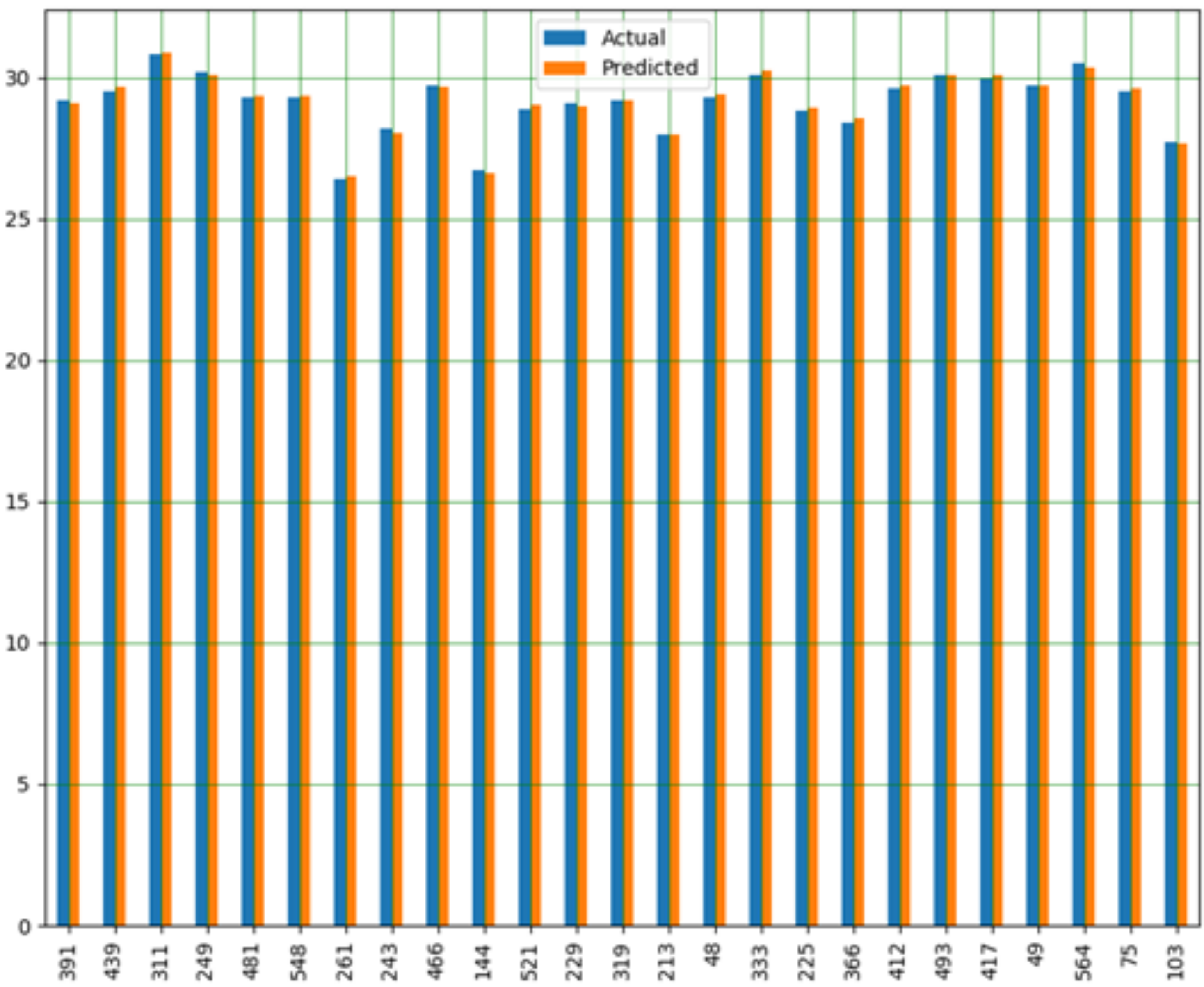






	Day	Month	Year	mean_temp	max_temp	min_temp	meanhum	meandew	pressure	heat	wet
0	1	8	1999	30.0	32.6	28.3	77	25.5	1002.1	YES	NO
1	2	8	1999	29.8	32.6	27.4	78	25.4	999.1	NO	NO
2	3	8	1999	29.7	31.4	28.3	80	25.8	998.4	NO	YES
3	4	8	1999	28.3	30.3	25.7	87	25.8	999.6	NO	YES
4	5	8	1999	28.4	32.6	25.3	82	24.9	999.6	NO	YES

Day	Month	Year	Heat	Wet
1	8	2019	NO	YES
2	8	2019	NO	YES
3	8	2019	NO	YES
4	8	2019	NO	YES
5	8	2019	NO	NO
6	8	2019	NO	NO
7	8	2019	YES	NO
8	8	2019	YES	NO



Conclusion and Future work

Due to lack of time and lack of hardware the whole project was not as complete as I first imagined to do while starting the project. The basic goal and underling technology are together and working.

I can me good UI for the user and some more research on getting good data will help model to perform even better.

References

- Salonidis, T., Bhagwat, P., & Tassiulas, L. (2000, August). Proximity awareness and fast connection establishment in Bluetooth. In 2000 First Annual Workshop on Mobile and Ad Hoc Networking and Computing. MobiHOC (Cat. No. 00EX444) (pp. 141-142). IEEE.

- Parashar, A. (2019, September). IoT based automated weather report generation and prediction using machine learning. In 2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT) (pp. 339-344). IEEE.

- Dutta, K., & Gouthaman, P. (2020). Rainfall prediction using machine learning and neural network. International Journal of Recent Technology and Engineering (IJRTE), 9(1), 1954-1961.

- Chao, Z., Pu, F., Yin, Y., Han, B., & Chen, X. (2018). Research on real-time local rainfall prediction based on MEMS sensors. Journal of Sensors, 2018.

- Su, C., Hou, D., Zhao, S., Li, X., Yuan, Z., & Yu, D. (2020). Remote control rain sampler for rainfall runoff collection. Water Supply, 20(2), 644-651.

Code:

```
#!/usr/bin/env python
# coding: utf-8

# In[81]:


from bs4 import BeautifulSoup
import urllib.request
from time import sleep
from datetime import datetime
import pandas as pd
import requests
import re
from datetime import date
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb


# In[82]:


def getweather():
    weather = []
    year = date.today().year
    url = "http://www.hko.gov.hk/cis/dailyExtract/
dailyExtract_" + str(year) + "08.xml"

    page = requests.get(url)

    soup = BeautifulSoup(page.content, 'lxml')

    body = soup.find("body").text
    body = body.split(",")
    # weather.append(temp)
    day = body[-27][4:-1]
    if (day[0] == "0"):
        day_2 = day[-1]
    else:
        day_2 = day
    weather.append(day_2)
    month = body[0][-1]
    weather.append(month)
    weather.append(year)

    temp = body[-24][1:-1]
    weather.append(temp)

    high = body[-25][1:-1]

    low = body[-23][1:-1]
    weather.append(high)
    weather.append(low)

    Humidity = body[-21][1:-1]
    weather.append(Humidity)
    dew = body[-22][1:-1]
    weather.append(dew)
```

```
    pressure = body[-26][1:-1]
    weather.append(pressure)
    if (float(temp) >= 30):
        heat = "YES"
    else:
        heat = "NO"
    weather.append(heat)
    if (float(Humidity) >= 80):
        wet = "YES"
    else:
        wet = "NO"
    weather.append(wet)

    print("Day : ", day_2)
    print("Month : ", body[0][-1])
    print("Year : ", year)
    print("Pressure : ", body[-26][1:-1])
    print("Max Temp : ", body[-25][1:-1])
    print("Mean Temp : ", body[-24][1:-1])
    print("Min Temp : ", body[-23][1:-1])
    print("Dew Point : ", body[-22][1:-1])
    print("Humidity : ", body[-21][1:-1])
    print("Heat : ", heat)
    print("Wet : ", wet)
    # print("Temperature : ",temp)

    return weather

# In[83]:


getweather()


# In[84]:


import csv
import openpyxl
import sys

def main():
    print("Collecting  Weather Data in a intervals of 30
mins")
    idx = 0
    weatherdata = {'Day':[],'Month':[],'Year':[],'mean_temp':
[],'max_temp':[],'min_temp':[],'meanhum':[],'meandew':
[],'pressure':[],'heat':[],'wet':[]}
    filename = 'data.xlsx'
    wb = openpyxl.load_workbook(filename=filename)
    sheet = wb['Sheet1']
    #new_row =
['Day','Month','Year','mean_temp','max_temp','min_temp','meanh
um','meandew','pressure','heat','wet']
    #sheet.append(new_row)
    while idx < 1:
        print('collecting weather data '+str(idx))
        tmp = getweather()
        weatherdata['Day'].append(tmp[0])
        weatherdata['Month'].append(tmp[1])
        weatherdata['Year'].append(tmp[2])
        weatherdata['mean_temp'].append(tmp[3])
        weatherdata['max_temp'].append(tmp[4])
        weatherdata['min_temp'].append(tmp[5])
        weatherdata['meanhum'].append(tmp[6])
        weatherdata['meandew'].append(tmp[7])
        weatherdata['pressure'].append(tmp[8])
```

```
        weatherdata['heat'].append(tmp[9])
        weatherdata['wet'].append(tmp[10])

        try :
            new_row =
[int(tmp[0]),int(tmp[1]),int(tmp[2]),float(tmp[3]),float(tmp[4
]),float(tmp[5]),float(tmp[6]),float(tmp[7]),float(tmp[8]),tmp
[9],tmp[10]]
            sheet.append(new_row)
        except ValueError as e:
            print (e)
        idx+=1

    wb.save(filename)


    print(weatherdata)


# In[85]:


main()


# In[86]:


import pandas as pd
data_xls = pd.read_excel('data.xlsx', 'Sheet1',
index_col=None)
data_xls.to_csv('data.csv', encoding='utf-8', index=False)


# In[87]:


#importing classes to handle data,to split data into training
and testing sets,to visualising of tree and to cheak accuracy
of our model
from preprocessing import splitter
from preprocessing import Encoder
from preprocessing import scaler
from metrics import matrix
from treeVis import vis
sp=splitter()
mt=matrix()
vs=vis()
sc=scaler()
en=Encoder()


# In[88]:


#importing dataset
dataset=pd.read_csv("data.csv")
dataset.tail()


# In[89]:


from sklearn.preprocessing import LabelEncoder
lab=LabelEncoder()
dataset.iloc[:,0]=lab.fit_transform(dataset.iloc[:,0])
dataset.iloc[:,1]=lab.fit_transform(dataset.iloc[:,1])
```

```
dataset.iloc[:,2]=lab.fit_transform(dataset.iloc[:,2])
dataset.iloc[:,3]=lab.fit_transform(dataset.iloc[:,3])
dataset.iloc[:,4]=lab.fit_transform(dataset.iloc[:,4])
dataset.iloc[:,5]=lab.fit_transform(dataset.iloc[:,5])
dataset.iloc[:,6]=lab.fit_transform(dataset.iloc[:,6])
dataset.iloc[:,7]=lab.fit_transform(dataset.iloc[:,7])
dataset.iloc[:,8]=lab.fit_transform(dataset.iloc[:,8])
dataset.iloc[:,9]=lab.fit_transform(dataset.iloc[:,9])
dataset.iloc[:,10]=lab.fit_transform(dataset.iloc[:,10])

dataset.tail()


# In[90]:


#deviding data into dependant and independant sets
x = dataset[['Day','Month','Year']]
y = dataset['heat']
z = dataset['wet']


# In[91]:


#visualising each attribute of dataset using histogram
dataset.hist(figsize = (30, 30))
plt.savefig("dataset.png")
plt.show()


# In[92]:


#deviding data into training and testing sets
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0
.20,random_state=42)
x_train,x_test,z_train,z_test=train_test_split(x,z,test_size=0
.20,random_state=42)


# In[93]:


#importing decision tree model and fitting training data to it
from sklearn.tree import DecisionTreeClassifier
classifier=DecisionTreeClassifier(criterion='entropy',random_s
tate=0)
classifier.fit(x,y)

classifier_z=DecisionTreeClassifier(criterion='entropy',random
_state=0)
classifier_z.fit(x,z)


# In[94]:


#predicting values for x_test and compairing result with
y_test
print("Heat:-")
y_pred=classifier.predict(x_test)
print("predicted values:-")
print(y_pred.astype(int))
y_test_arr=np.array(y_test)
print("original values:-")
```

```
print(y_test_arr)

#predicting values for x_test and compairing result with
y_test
print("Wet:-")
z_pred=classifier_z.predict(x_test)
print("predicted values:-")
print(z_pred.astype(int))
z_test_arr=np.array(z_test)
print("original values:-")
print(z_test_arr)


# In[95]:


#cheaking accuracy of our model
accuracy=mt.accuracy(y_pred,y_test)
print("{}%".format(accuracy*100))

accuracy=mt.accuracy(z_pred,z_test)
print("{}%".format(accuracy*100))


# In[96]:




# In[99]:


from datetime import date
now = date.today()
filepath = "prediction/forecast-"+str(now)+".xlsx"
def prediction():
    wb = openpyxl.Workbook()
    wb.save(filepath)
    print("Predict Weather Data ")
    idx = 0
    weatherdata = {'Day':[],'Month':[],'Year':[],'heat':
[],'wet':[]}
    filename = filepath
    wb = openpyxl.load_workbook(filename=filename)
    sheet = wb['Sheet']
    new_row = ['Day','Month','Year','Heat','Wet']
    sheet.append(new_row)
    while idx < 31:
        print('Predict weather data ' + str(idx))
        year = date.today().year
        yr = year - 1999
        if (date.today().day == 31):
            year += 1
            yr += 1
        y_pred = classifier.predict([[idx, 0, yr]])
        print(y_pred.astype(int))
        z_pred = classifier_z.predict([[idx, 0, yr]])
        print(z_pred.astype(int))
        tmp_d = idx + 1
        tmp_m = 8
        tmp_y = year
        tmp_h = y_pred.astype(int)
        tmp_w = z_pred.astype(int)
        if (int(tmp_h) == 0):
            tmp_h_char = "NO"
        else:
            tmp_h_char = "YES"
```

```
        if (int(tmp_w) == 0):
            tmp_w_char = "NO"
        else:
            tmp_w_char = "YES"
        new_row = [int(tmp_d), int(tmp_m), int(tmp_y),
tmp_h_char, tmp_w_char]
        sheet.append(new_row)
        idx += 1
    wb.save(filename)

    print(weatherdata)


# In[100]:


prediction()


# In[101]:


data_xls = pd.read_excel(filepath, 'Sheet', index_col=None)
data_xls.to_csv("prediction/forecast-"+str(now)+".csv",
encoding='utf-8', index=False)


# In[102]:


#importing dataset
data_pred=pd.read_csv("prediction/forecast-"+str(now)+".csv")
data_pred


# In[103]:


import seaborn as sns
sns_plot = sns.countplot(x='Heat',data=data_pred)
fig = sns_plot.get_figure()
fig.savefig("prediction/heat_summary_prediction-"+str(now)
+".png")


# In[104]:


sns_plot = sns.countplot(x='Wet',data=data_pred)
fig = sns_plot.get_figure()
fig.savefig("prediction/wet_summary_prediction-"+str(now)
+".png")
```

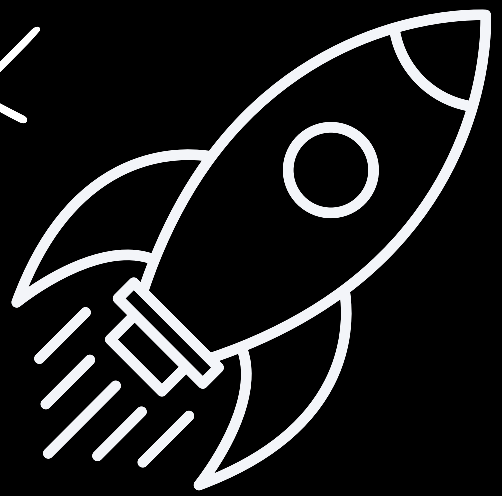
VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

REVERSE CLOUD SEEDING

USING ROCKETS



NAVDEEP SUREKA
19BCE2679

PROBLEM STATEMENT

To develop the reverse cloud seeding and rain monitoring system using IOT control devices to effectively predict the rain using weather prediction and launch the planned payload to control the rainfall in the particular area. The IoT system will allow different rockets to communicate and initiate the launch sequence and also provide real-time location of the rocket after launch for a successful recovery..

CLOUD SEEDING

TITLE : CLOUD SEEDING USING ROCKETS

CLOUD SEEDING ALTERS THE MOISTURE AVAILABLE IN A CLOUD TO CREATE ADDITIONAL RAINFALL. THIS IS DONE BY DISPERSING SILVER IODIDE OR SODIUM SALTS INTO THE CLOUDS AND AFFECTS THE PHYSICAL PROCESSES OCCURRING INSIDE IT. SILVER IODIDE IS AFFECTIVE BECAUSE IT IS SIMILAR TO ICE CRYSTALS. THESE SUBSTANCES CAN BE EASILY DELIVERED INTO THE CLOUDS BY OUR GROUND-BASED ROCKET. THESE SUBSTANCES ATTRACT WATER VAPOR TO THE CLOUD AND MAKE LARGER WATER DROPLETS WHICH IN TURN MAKE THE CLOUD HEAVY AND INCREASE THE CHANCE OF RAIN TO A VERY HIGH EXTENT.

ADVANTAGES

One of the only paper in this domain and the idea are well tested.

DISADVANTAGES

The chemical used for the cloud seeding is hard to get hand-on.

DATE Jun 6, 2020

COMMUNICATION ESTABLISHMENT

TITLE : PROXIMITY AWARENESS AND FAST CONNECTION ESTABLISHMENT IN BLUETOOTH

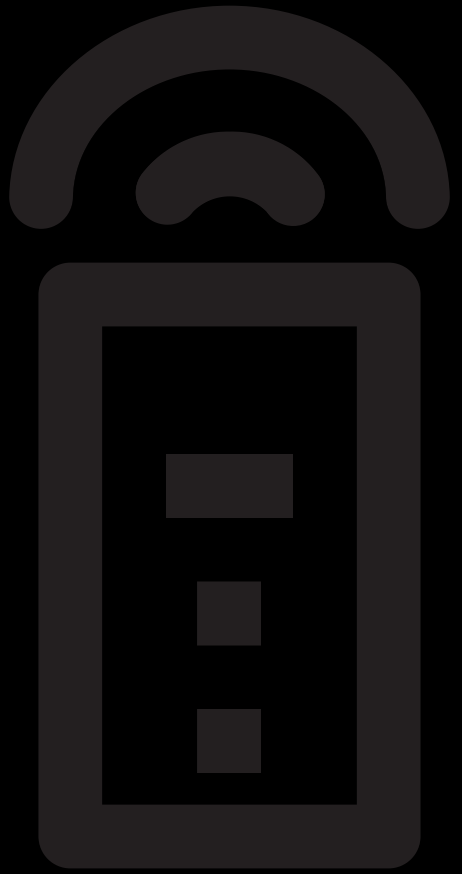
A SYMMETRIC TECHNIQUE FOR ESTABLISHING AD HOC CONNECTIVITY IS INTRODUCED WHICH IMPOSES EACH NODE TO ALTERNATE BETWEEN THE "SENDER" AND "RECEIVER" STATE IN A RANDOM FASHION.

ADVANTAGES

Easy to implement as the chip used is very affordable and the reliability is good enough.

DISADVANTAGES

Have to set a time out for all devices as the comunication is not parallel.



DATE was not dated

CLOUD STORAGE

TITLE : RESEARCH ON CLOUD DATA STORAGE TECHNOLOGY AND ITS ARCHITECTURE IMPLEMENTATION

ANALYZES THE CLOUD DATA STORAGE TECHNOLOGY GFS(GOOGLE FILE SYSTEM)/HDFS(HADOOP DISTRIBUTED FILE SYSTEM)

ADVANTAGES

The data processing is fast and we can use google cloud to perform analyzes without additional code .

DISADVANTAGES

Based on google cloud and using the system with other provider is hard.

DATE 2018

WEATHER PREDICTION

TITLE : IOT BASED AUTOMATED WEATHER REPORT GENERATION AND PREDICTION USING MACHINE LEARNING

THE MODEL WILL SENSE DIFFERENT WEATHER PARAMETERS LIKE PRESSURE, HUMIDITY, RAINFALL, TEMPERATURE, DUST PARTICLES AND LIGHT ETC. AND WILL DISPLAY CURRENT DAY'S TEMPERATURE ON A WEB
PORTAL. THE MODEL WILL ALSO BE ABLE TO PREDICT THE MAXIMUM
AND MINIMUM TEMPERATURES OF THE NEXT DAY USING PAST THREE
DAYS' DATA.

ADVANTAGES

It used both ground based data and data from weather povider and based on both data, it gives a good balance of weather.

DISADVANTAGES

But for our use, the system will be very much bias on the online prediction as on-ground sensors are not very powerful.

DATE Sep 28-29, 2019

GPS TRACKING

TITLE : IMPLEMENTATION OF GPS BASED OBJECT LOCATION AND ROUTE TRACKING ON ANDROID DEVICE

THIS PROJECT DEVELOPS A TRACKING / MONITORING ANDROID APPLICATION (MOBILE) USING OBJECT GPS DEVICES TO ASCERTAIN ITS CURRENT LOCATION, AND PREVIOUS LOCATION AT SPECIFIED INTERVALS, THIS SYSTEM UNLIKE PREVIOUS TRACKING SYSTEM WILL GIVE USER THE ABILITY TO CREATE BOOKMARK OF CURRENT LOCATION AND ABILITY TO ROUTE BACK TO THAT LOCATION FROM ANYWHERE USING GOOGLE MAPS API'S IN CASE THEY CAN'T REMEMBER THE PRICES LOCATION.

ADVANTAGES

The system have an tracking based on android endpoints so we can easily implement it using simple API call.

DISADVANTAGES

Not used for arial object tracking and the hard impact is not yet tested.

DATE November, 2015

RAIN PREDICTION

TITLE: RAINFALL PREDICTION USING MACHINE LEARNING AND NEURAL NETWORK

THERE ARE TWO TECHNIQUES TO PREDICT RAINFALL. THE FIRST ONE IS MACHINE LEARNING APPROACH, WHICH INCLUDES LASSO REGRESSION. THE SECOND ONE IS NEURAL NETWORK APPROACH. THIS SYSTEM FIRST COMPARES BOTH THE PROCESS AND THEN ACCORDINGLY GIVES RESULT WITH THE BEST ALGORITHM.

ADVANTAGES

As it uses Lasso regression , the data produced can be very much compressed and can be easily communicated.

DISADVANTAGES

The implementatin of code is not avaliable and the reader have to code the stuff based on the theory.

DATE 1, May 2020

RAIN PREDICTION

TITLE:RESEARCH ON REAL-TIME LOCAL RAINFALL PREDICTION BASED ON MEMS SENSORS

SEASONAL TREND DECOMPOSITION USING LOESS (STL) ALGORITHM IS UTILIZED TO DECOMPOSE THE OBSERVED
TIME SERIES INTO TREND, SEASONAL, AND REMAINDER COMPONENTS.

SECONDLY, THE LONG SHORT-TERM MEMORY (LSTM) IS USED TO PREDICT THE REAL-TIME RAINFALL BASED ON THE OBSERVED DATA.

ADVANTAGES

This paper provides a good balance between short term prediction and long term prediction.

DISADVANTAGES

The implementatin of code is not avaliable and the reader have to code the stuff based on the theory.

DATE 8 February 2018

REMOTE CONTROL

TITLE :REMOTE CONTROL RAIN SAMPLER FOR RAINFALL RUNOFF COLLECTION

ACCORDING TO THE WHOLE SYSTEM, WHEN THE TRIGGER DEVICE DETECTS RAINFALL RUNOFF, AN ENERGY-SAVING CIRCUIT CONNECTS, AND THE SAMPLER IS POWERED. AT THE SAME TIME, STATE AND POSITION INFORMATION ARE SENT TO THE MOBILE PHONE THROUGH THE COMMUNICATION MODULE. THEN, THE MOBILE PHONE SENDS INSTRUCTIONS AND STARTS AN ORDER BACK TO THE SAMPLER THROUGH TEXTS BASED ON DIFFERENT REQUIREMENTS.
THE INSTRUCTIONS AND START ORDER ARE THEN PASSED TO THE SINGLE-CHIP MICYOCO (SCM) CONTROL SYSTEM, WHICH ALSO OCCURS VIA THE COMMUNICATION MODULE. A DISCONNECTED SIGNAL WILL BE SENT TO THE ENERGYSAVING CIRCUIT AFTER THE RAINFALL RUNOFF COLLECTION.

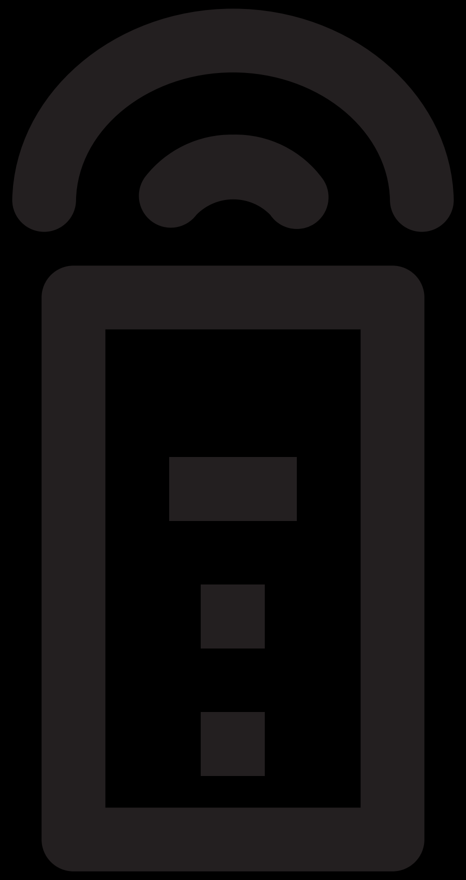
ADVANTAGES

With the energy-saving circuit, the standby time of sampler is high compared with no energy-saving circuit.

DISADVANTAGES

Achieving online detection of rainfall runoff water samples on the existing basis is not possible using this system.

DATE 2020



GPS TRACKING

TITLE : DEVELOPING A PASSIVE GPS TRACKING SYSTEM TO STUDY LONG-TERM TRAVEL BEHAVIOR

THE ALGORITHMS USED TO DEVELOP THE GPS SYSTEM CONSISTED OF:

- ACTIVITY AND TRIP IDENTIFICATION(DIVIDING THE USERS' RECORDS INTO ACTIVITIES AND TRIPS);
- TRIP SEGMENTATION (GROUPING TRIPS INTO WALKING OR USING SOME MEANS OF TRANSPORT);
- AND, MODE DETECTION (IDENTIFYING THE TRANSPORT MEANS).

ADVANTAGES

Advantage was of using users' past data to improve mode detection results.

DISADVANTAGES

Not used for arial object tracking and the hard impact is not yet tested.

DATE 31 October 2018

CLOUD STORAGE

TITLE : RESEARCH ON DATA STORAGE TECHNOLOGY IN CLOUD COMPUTING ENVIRONMENT

DATA BLOCKING ALGORITHM IS STUDIED TO SHORTEN THE RESPONSE TIME OF DATA BACKUP.

ADVANTAGES

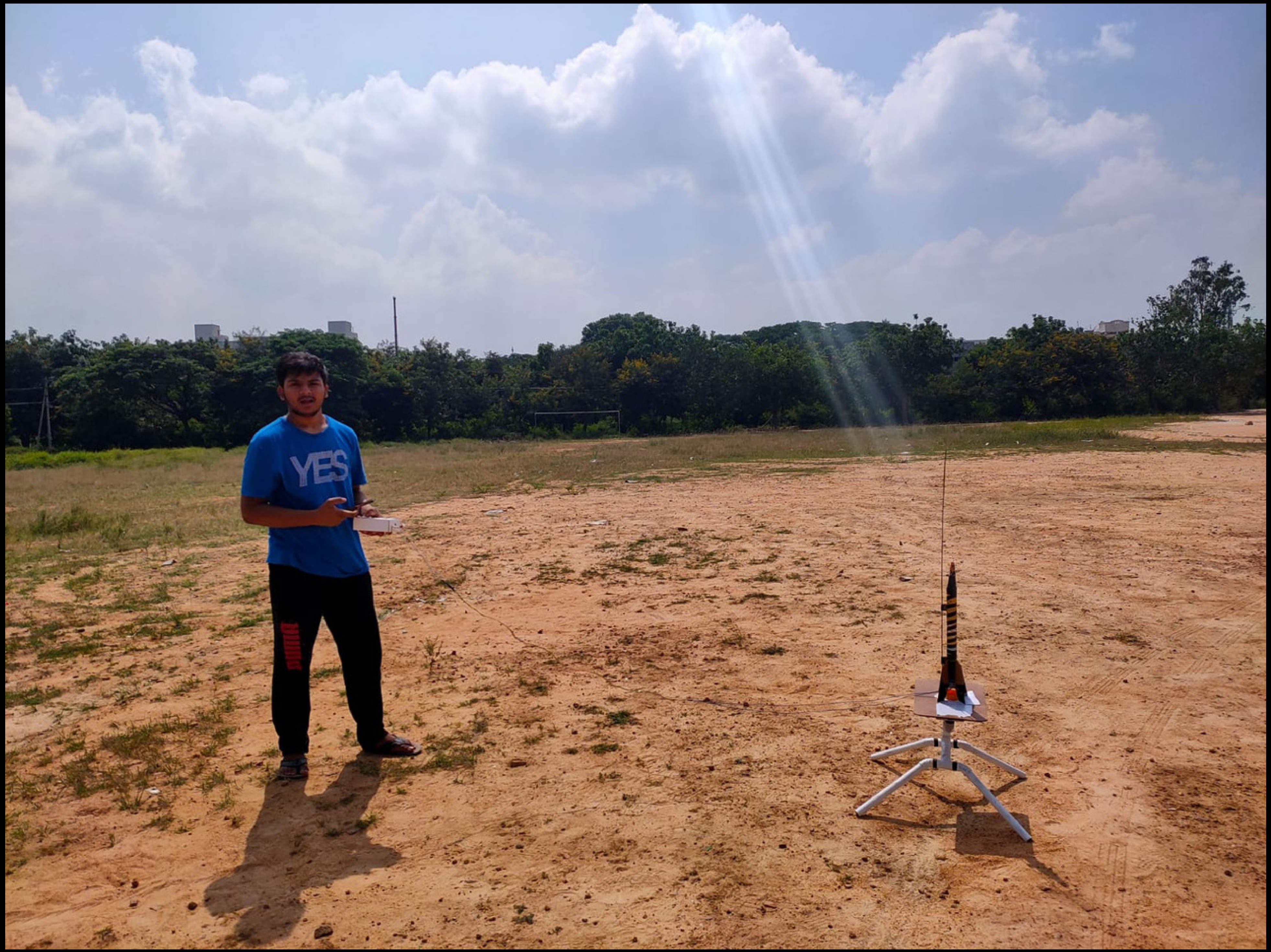
The advantages of the fixed-length block algorithm are simple and high performance, but it is very sensitive to data insertion and deletion.

DISADVANTAGES

Is very inefficient and cannot be adjusted and optimized according to the content changes.

DATE 2018

IMAGES



Proposed idea

Use a ground-based weather monitoring station and based on data from the ground instrument and weather forecast prediction, our model will give a better rain prediction and using the data we can calculate the payload capacity.

Tools- Software/Hardware required

We will be using an open-source application and data points.

HARDWARE

- HC-05 Bluetooth module
- Microcontroller
- DHT 11 sensor
- Rain drop sensor
- ESP8266 wifi mod

SOFTWARE / TOOLS

- Node.js with express
- Python3 with sklearn
- Flask with express
- AWS - if we want cloud integration
- weatherstack api
- Jupyter notebook
- MongoDB

Video Viewer

1080px(W) 1920px(H)

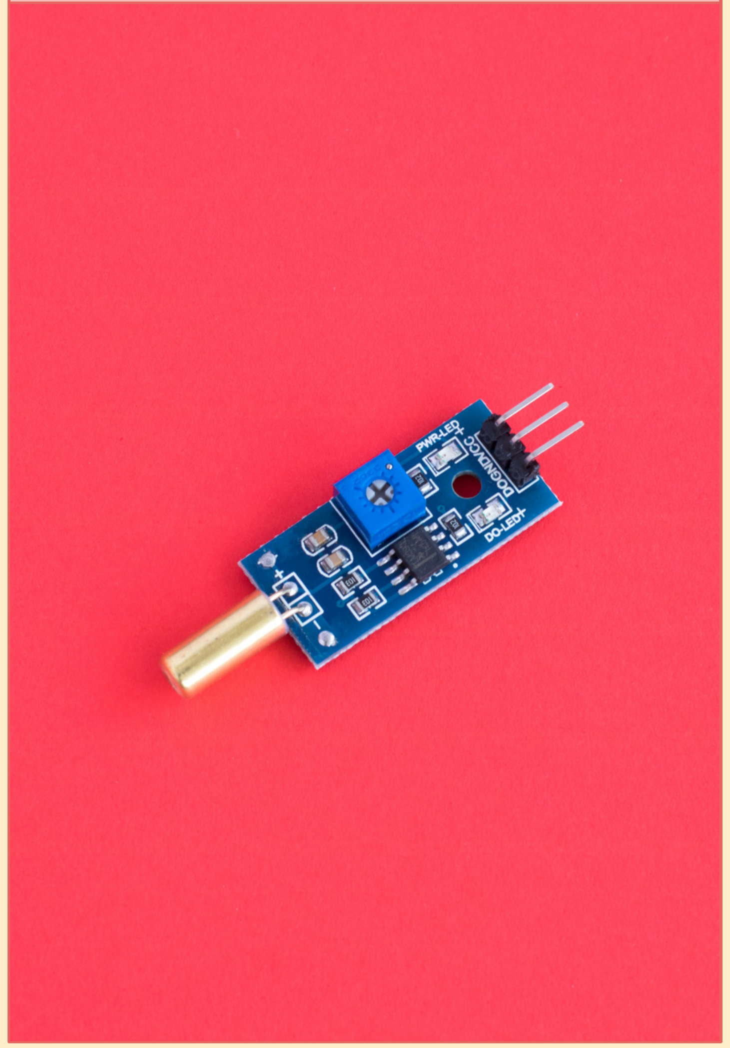
A three-step process

watch_this.mp4

Workflow - Step by Step procedure

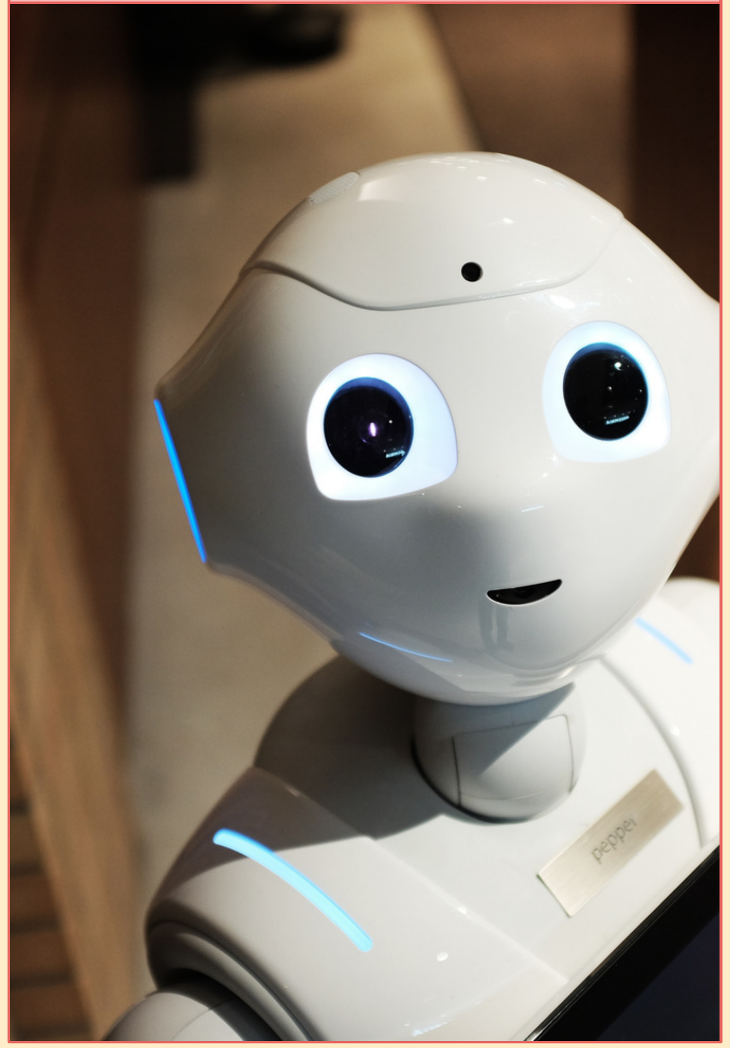
Historical data

Historical data from the weatherstack API using a node-express app



Sensor data

Data from ground-based sensor.



ML Algorithm

Based on both data, ML algo will make the rain predictions and bootstrap the rockets.

WorkFlow

1 **Create the API**

Make the weatherstack API as a standalone application

2 **Create data collection**

System to store and manage the data from the ground-based hardware.

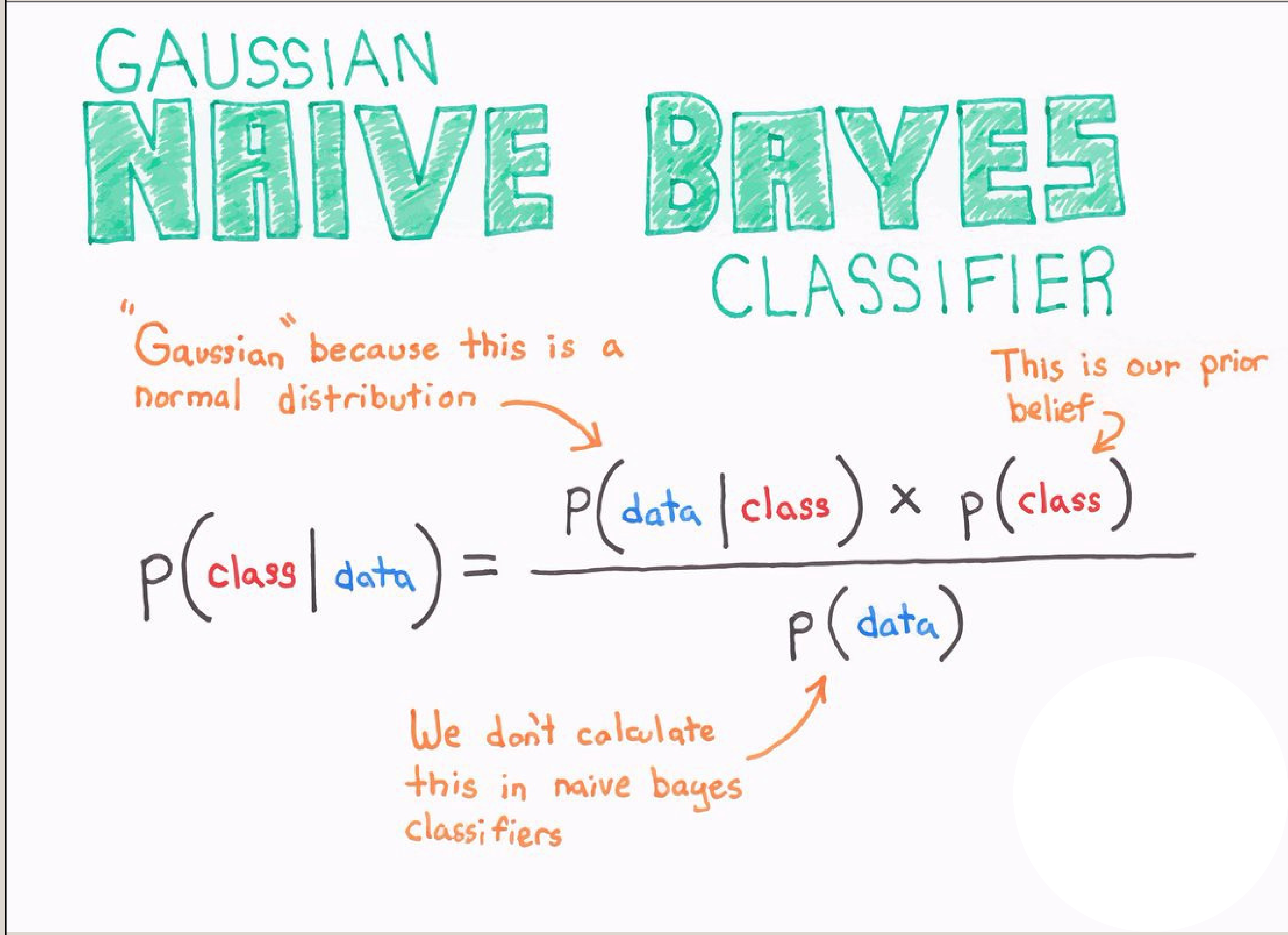
3 **Machine learning**

Using flask to host pre-trained ML model

4 **Configuring rockets**

Based on the predictions, data will be sent to rockets to auto-configure

Algorithm



Using Gaussian classifier in Naive Bayes for weather prediction

Library

sklearn.naive_bayes.GaussianNB

Reason

Best for prediction probability of events based on historical data

Improvement

Backfeed new information and update the base model