CSE1901 - Technical Answers to Real World Problems (TARP)

Project Report

Impact of Weather on Agriculture

By

19BCE1864	Maulishree Awasthi
19BCE1871	Rekha V

B. Tech Computer Science and Engineering

Submitted to

Renukadevi Saravanan

School of Computer Science and Engineering

April 2022

DECLARATION

I hereby declare that the report titled "**Impact of weather on agriculture"** submitted by Maulishree Awasthi and Rekha V to VIT Chennai is a record of bona-fide work undertaken by us under the supervision of **Renukadevi Saravanan**, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai.

Maulishree Awasthi

Reg. No. 19BCE1864

Rekha V

Reg. No. 19BCE1871

CERTIFICATE

Certified that this project report entitled "**Impact of Weather on Agriculture"** is a bona-fide work of **Maulishree Awasthi (19BCE1864), Rekha V (19BCE1871)** and they carried out the Project work under my supervision and guidance for CSE1901 - Technical Answers to Real World Problems (TARP).

Renukadevi Saravanan
SCOPE, VIT Chennai

ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our project guide, Dr. Renukadevi Saravanan, Associate Professor, School of Computer Science Engineering, for her consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We are extremely grateful to Dr. Ganesan R, Dean of School of Computer Science Engineering, VIT Chennai, for extending the facilities of the school towards our project and for her unstinting support.

We express our thanks to our Head of the Department Dr. Nithyanandam P, for his support throughout the course of this project. We also take this opportunity to thank all the faculty of the school for their support and their wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

Maulishree Awasthi   Rekha V
19BCE1864   19BCE1871

ABSTRACT

A major population of any country across the globe depends on agriculture for survival. Many times, resources are wasted when the farmers are not able to correctly predict the crop to be grown due to lack of information. We have built a model that predicts the type of crop that can be cultivated in a given land on the basis of climatic conditions. Therefore, the crop with maximum production can be obtained. This project was developed because to help user access data about weather anywhere in real-time. This paper proposes a viable and user-friendly yield prediction system for the farmers. The proposed system collects data from IOT based sensors and stores it in cloud. The real time weather conditions are fed to a machine learning model which in turn predicts the crop based on the parameters and the soil conditions. A mobile application provides an easy and user-friendly interface which helps user to interact with the model and monitor the weather parameters like temperature, pressure and humidity in real time. The application alerts the user when any of the mentioned parameters exceeds the safe limit for crop growing. An additional feature of the application is to monitor the life cycle of the crop from the sowing of seeds to the harvest of a healthy full-grown crop.

CONTENTS

INTRODUCTION

1.1 OBJECTIVE AND GOAL OF THE PROJECT

Agriculture is the main occupation of our country engaging more than 50% of Indian work force. India is the second largest producer of wheat and rice in the world. However lack of knowledge regarding climatic conditions and soil results in poor productivity of the crops. Crop selection and changing climatic conditions are the two major problems faced among the farmers. Each crop has its own suitable climatic features. This can be handled with the help of precise farming techniques. With the help of IOT and Machine Learning these problems can be tackled efficiently. The system built in this project uses a perfect blend of IOT and Machine Learning to improve the crop production by studying the soil nutrients and climatic conditions. In addition to its basic functionality of predicting the crop, the project also focuses on real time monitoring of weather elements such as temperature, pressure and humidity along with their visual representation. The system also provides an easy tracking of the crop life cycle from sowing the seeds till the harvest of the crop giving periodic reminders regarding pesticides and irrigation of the crops. The system provides an easy user interface which helps user to interact with the model conveniently.

The project aims at tackling problems of the agriculture sector with the efficient use of modern scientific methods, thus increasing the GDP of the country.

1.2 PROBLEM STATEMENT

Agriculture has an extensive history in India. Agriculture is the essential wellspring of business for a major population of any country. The crop yield depends on multiple factors such as climatic, geographic, organic, and financial elements. It is difficult for farmers to decide when and which crops to plant because of fluctuating market prices. Farmers are unaware of which crop to grow, and what is the right time and place to start due to uncertainty in climatic conditions. The usage of various fertilizers is also uncertain due to changes in seasonal climatic conditions and basic assets such as soil, water, and air. In this scenario, the crop yield rate is steadily declining. The crop yield prediction is a significant problem in the agriculture sector. In this paper, we have proposed a model that addresses these issues. The novelty of the proposed system is to guide the farmers to maximize the crop yield as well as suggest the most profitable crop for the specific region. The proposed model provides crop selection based on economic and environmental conditions, and benefit to maximize the crop yield.

This project collects information about the environmental conditions. Firstly, the temperature, humidity and pressure data are collected from the environment. The data are collected using IOT sensors. Firstly, the software simulation is done circuit is done in Proteus Professional. Next the same design is implemented in the hardware. The sensors are connected with Arduino-UNO which acts as an interface between the sensors and the GSM module. The GSM module is used to sends the collected data to the Thingspeak platform using the cellular data. The GSM uses a TCP connection to write data to the Thingspeak. In the Thingspeak platform the data is collected in the JSON format. The collected data can be visualized using charts.

This system also provides a mobile application which used to monitor the temperature, humidity and pressure parameters of the environment. The mobile application also alerts the user if the parameters exceed the threshold temperature or humidity or pressure. The farmer is also given a permission to change the threshold values of the parameters. The mobile app also displays the graphs of real-time arrival of temperature, humidity and pressure.

A machine learning model is built to predict the weather and suggest what are suitable for growing in that particular weather. The parameter used for predicting the suitable crop is temperature, humidity, rainfall, pH, phosphorous, potassium and nitrogen. The crops which are used in the dataset are rice, maize, chickpea, kidney beans, moth beans, mung bean, black gram, lentil, pomegranate, banana, mango, grapes, watermelon, muskmelon, apple, orange, papaya, coconut, cotton, jute and coffee. Different types of machine learning models are used initially. The model with best accuracy is taken and used to predict the suitable crop. In the mobile application the farmer can also check which crop is suitable for growing in a particular area for the real-time data of temperature, humidity, pressure, pH value, phosphorous value of soil and potassium value of soil.

1.3 MOTIVATION

Agriculture has an extensive history in India. Recently, India is ranked second in the farm output worldwide. Agriculture-related industries such as forestry and fisheries contributed for 16.6% of 2009 GDP and around 50% of the total workforce. Agriculture's monetary contribution to India's GDP is decreasing. The crop yield is the significant factor contributing in agricultural monetary. The crop yield depends on multiple factors such as climatic, geographic, organic, and financial elements. It is difficult for farmers to decide when and which crops to plant because of fluctuating market prices. Citing to Wikipedia figures India's suicide rate ranges from 1.4-1.8% per 100,000 populations, over the last 10 years. Farmers are unaware of which crop to grow, and what is the right time and place to start due to uncertainty in climatic conditions. The usage of various fertilizers is also uncertain due to changes in seasonal climatic conditions and basic assets such as soil, water, and air. In this scenario, the crop yield rate is steadily declining. The solution to the problem is to provide a smart user-friendly recommender system to the farmers.

The crop yield prediction is a significant problem in the agriculture sector. Every farmer tries to know crop yield and whether it meets their expectations, thereby evaluating the previous experience of the farmer on the specific crop predict the yield. Agriculture yields rely primarily on weather conditions, pests, and preparation of harvesting operations. Accurate information on crop history is critical for making decisions on agriculture risk management.

1.4 CHALLENGES

Despite many solutions that have been recently proposed, there are still open challenges in creating a user-friendly application with respect to crop recommendation. The solution proposed here aims to solve these limitations, by developing a user-friendly application that considers the parameters like rainfall, temperature, soil type etc. that directly affect cultivation. The main objective is to obtain a better variety of crops that can be grown over the season. The proposed system would help to minimize the difficulties faced by farmers in choosing a crop and maximize the yield in effect to reduce the suicide rates.

LITERATURE SURVEY

[1] IoT Based Weather Monitoring System Using Arduino-UNO

In this paper they read the temperature and humidity sensor information. In this paper they are using a framework that screens the climate progressively over a portable application. For recognizing precipitation or snow fall, the raindrop sensors are used. In this paper they have also used LDR to display the parameter values like temperature, pressure and humidity. The microcontroller Arduino UNO further transmits the data recorded to Thingspeak using Wi-Fi module.

[2] IoT Based Environmental Monitoring System using Arduino UNO and Thingspeak

This paper talks about using sensors like temperature, humidity, rain-water level detector sensor, gas sensors and pressure sensor for monitoring the weather conditions. And they have also used Arduino UNO and WIFI module that helps in processing and transferring of data the Thingspeak cloud. The changes in the environment are updated in the form of database through the cloud computing method. In this paper Thing speak is also used to provide a feature to create a public based channel to analyze and estimate it through the public. An Android application is created for the direct access of the measured parameters. In this paper the measured parameters from the sensors are continuously updated and is thus viewed by the user using the EMS (Environmental Monitoring System) application. Therefore, the data is directly accessed and is purely independent of third parties.

[3] Real Time Weather Prediction System Using IOT and Machine Learning

This paper implements the real time weather prediction system that can be used in number of applications like homes, industries, agriculture, stadiums etc. for predicting the weather information. The system utilizes a temperature and humidity sensor i.e., DHT11 and a light intensity sensor i.e., LDR. The sensed data from the sensors are uploaded to a Thingspeak cloud server using NodeMCU and ESP8266-01 module. The data is also displayed on a customized HTML webpage for monitoring the real time values. A logistic regression model is used for setting up the machine learning environment. This model is trained using the pre- recorded values of sensor data. This real time data is used to test the model and prediction is done for a particular value by blinking the led connected to NodeMCU.

[4] Crop Recommender System Using Machine Learning Approach

This paper proposes a viable and user-friendly yield prediction system for the farmers. The proposed system provides connectivity to farmers via a mobile application. GPS helps to identify the user location. The user provides the area & soil type as input. Machine learning algorithms allow choosing the most profitable crop list or predicting the crop yield for a user-selected crop. To predict the crop yield, selected Machine Learning algorithms such as Support Vector Machine (SVM), Artificial Neural Network (ANN), Random Forest (RF), Multivariate Linear Regression (MLR), and K-Nearest Neighbors (KNN) are used. Among them, the Random Forest showed the best results with 95% accuracy. Additionally, the system also suggests the best time to use the fertilizers to boost up the yield.

[5] Crop Recommendation System using Machine Learning

The proposed system of IoT and ML is enabled for soil testing using the sensors, is based on measuring and observing soil parameters. This system lowers the probability of soil degradation and helps maintain crop health. Different sensors such as soil temperature, soil moisture, pH, NPK, are used in this system for monitoring temperature, humidity, soil moisture, and soil pH along with NPK nutrients of the soil respectively. The data sensed by these sensors is stored on the microcontroller and analyzed using machine learning algorithms like random forest based on which suggestions for the growth of the suitable crop are made. This project also has a methodology that focuses on using a convolutional neural network as a primary way of identifying if the plant is at risk of a disease or not.

[6] Improvement of Crop Production Using Recommender System by Weather Forecasts

Establishing linkages between Meteorological and climatic data, and farming decision-making is a challenging task. The following paper addresses the challenges associated with this. This paper presents a predictive analysis to analyses the best crop which can be produced for specific weather conditions and also suggests a hybrid recommender system that adopts CBR - Case-Based Reasoning for enhancing the success ratio of the system. This proposed novel hybrid system is a combination of the collaborative filtering technique and case-based reasoning. The novelty of the model lies in the of district-wise agriculture data analysis for predicting future climatic conditions and recommending crops based on those climatic conditions and also considering the agriculture pattern of the district using a hybrid recommender system.

REQUIREMENTS SPECIFICATION

3.1 HARDWARE REQUIREMENTS

The proposed used consists of the following hardware components:

- Arduino Uno
- BMP – 180 sensors
- DHT11 sensor
- GSM module
- LCD display
- Connecting Wires
- SIM card

GSM module is used to connect to the cloud i.e., Thingspeak.

3.2 SOFTWARE REQUIREMENTS

In the software simulation the design of the circuit is constructed using Proteus Professional software. The coding for the hardware implementation is done using Arduino software.

The mobile application is implemented using:

- Android Studio (Java language)

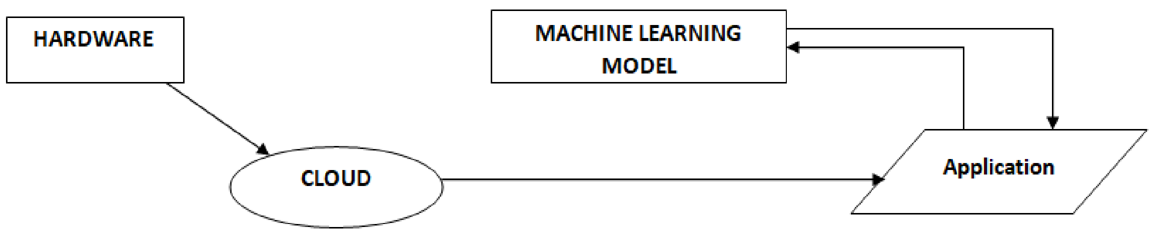
The machine learning model is built using:

- Python

Some of the libraries handy in building the project are numpy, sklearn, pandas and pickle-mixin. The model was finally converted into an API using flask module

SYSTEM DESIGN

The proposed system has both hardware and software components. The hardware components (sensors) collect real time data and store it in cloud. The software component collects the stored data and processes it to make valid predictions. A bird view of the project is as follows:



1. **Machine Learning model:** Different machine learning models are to be used and compared like:

   - ✓ Linear Regression
   - ✓ Logistic Regression
   - ✓ Random Forest
   - ✓ Decision Tree Classifier
   - ✓ MLP Classifier
   - ✓ Gradient Booster Classifier
   - ✓ Support Vector Classifier

   Python has been used to build the model with the help of its various libraries such as numpy, sklearn, pandas, pickle-mixin, etc.

2. **Hardware:** IOT based sensors BMP-180 AND DHT11 have been used to collect weather data in realtime. Arduino UNO is used to simulate the circuit.

3. **Cloud:** ThingSpeak Cloud Platform has been used to store the real time data collected from the sensors.

4. **Application:** The android application has been built using Native Java with an easy to use user interface and authenticated logins.

IMPLEMENTATION OF THE PROJECT

HARDWARE

The hardware built for the project aims at collecting real time weather components like temperature, humidity and pressure with the help of DHT11 AND BMP-180 sensors. The collected data is stored in ThingSpeak Cloud Platform from where it could be used later for processing and predictions. The hardware components used are:

- Arduino Uno
- BMP – 180 sensors
- DHT11 sensor
- GSM module
- LCD display
- Connecting Wires
- SIM card

The circuit diagram of the project is shown below:

SOFTWARE

Software consists of the two components:

a. **Machine Learning Model**

The system aims at fetching the local weather and soil parameters and predicting the crop best suited for the conditions provided. Different models were tested in order to find the best fitted model for the problem statement. The different machine learning models used are

i. **Linear Regression**

```
lr = LinearRegression().fit(X_train, y_train)
lr_pred= lr.score(X_test, y_test)

print("Training score: {:.3f}".format(lr.score(X_train, y_train)))
print("Test score: {:.3f}".format(lr.score(X_test, y_test)))
```

```
Training score: 0.303
Test score: 0.276
```

ii. **Decision Tree Classifier**

```
[ ] tree = DecisionTreeClassifier(max_depth=15,random_state=0).fit(X_train, y_train)
    tree_pred= tree.score(X_test, y_test)

    print("Training score: {:.3f}".format(tree.score(X_train, y_train)))
    print("Test score: {:.3f}".format(tree.score(X_test, y_test)))
```

```
Training score: 1.000
Test score: 0.987
```

iii. **Support Vector Classifier**

```
[ ] svm = SVC(C=100, gamma=0.001).fit(X_train, y_train)
    svm_pred= svm.score(X_test, y_test)

    print("Training score: {:.3f}".format(svm.score(X_train, y_train)))
    print("Test score: {:.3f}".format(svm.score(X_test, y_test)))
```

```
Training score: 0.998
Test score: 0.982
```

iv. Random Forests

```
[ ] rf = RandomForestClassifier(n_estimators=10, max_features=3, random_state=0).fit(X_train, y_train)
    rf_pred= rf.score(X_test, y_test)

    print("Training score: {:.3f}".format(rf.score(X_train, y_train)))
    print("Test score: {:.3f}".format(rf.score(X_test, y_test)))
```

```
Training score: 1.000
Test score: 0.989
```

v. Gradient Booster Classifier

```
[ ] gbr = GradientBoostingClassifier(n_estimators=20, max_depth=4, max_features=2, random_state=0).fit(X_train, y_train)
    gbr_pred= gbr.score(X_test, y_test)

    print("Training score: {:.3f}".format(gbr.score(X_train, y_train)))
    print("Test score: {:.3f}".format(gbr.score(X_test, y_test)))
```

```
Training score: 1.000
Test score: 0.982
```

vi. Logistic Regression

```
[ ] log_reg = LogisticRegression(C=0.1, max_iter=100000).fit(X_train, y_train)
    log_reg_pred= log_reg.score(X_test, y_test)

    print("Training score: {:.3f}".format(log_reg.score(X_train, y_train)))
    print("Test score: {:.3f}".format(log_reg.score(X_test, y_test)))
```

```
Training score: 0.993
Test score: 0.973
```

vii. MLP Classifier

```
[ ] # compute the mean value per feature on the training set
    mean_on_train = X_train.mean(axis=0)
    # compute the standard deviation of each feature on the training set
    std_on_train = X_train.std(axis=0)
    # subtract the mean, and scale by inverse standard deviation
    # afterward, mean=0 and std=1
    X_train_scaled = (X_train - mean_on_train) / std_on_train
    # use THE SAME transformation (using training mean and std) on the test set
    X_test_scaled = (X_test - mean_on_train) / std_on_train

    mlp = MLPClassifier(random_state=0, max_iter=10000, alpha=0.01,hidden_layer_sizes=[100,50])
    mlp.fit(X_train_scaled, y_train)

    mlp_pred= mlp.score(X_test_scaled, y_test)

    print("Training score: {:.3f}".format(mlp.score(X_train_scaled, y_train)))
    print("Test score: {:.3f}".format(mlp.score(X_test_scaled, y_test)))
```

```
Training score: 0.999
Test score: 0.975
```

On comparison of the above seven mentioned machine learning algorithms, random forest showed the best accuracy score of 98.9%. Therefore it was chosen to build the model.
Random forest is a supervised learning technique. It is a popular machine learning algorithm. It can be used for both Classification and Regression problems in Machine Learning. It is based on the process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. This concept is called ensemble learning. Random forest contains a number of decision trees. In order to improve the accuracy of the dataset predicted average is taken. Random forest takes the prediction from each tree rather than relying on one decision tree. Based on majority votes of predictions, it predicts the final output. The main advantage of Random forest is that it can handle large datasets with high dimensionality. It prevents over fitting issue by enhancing the accuracy of the model.

b. Android Application

In order to provide a user friendly easily accessible user interface for the system an android application has been built using standard JAVA programming language. The machine learning model built using Python is first converted into a subsequent API using flask module. The application collects data from user and sensors (cloud), calls the API, the API processes the inputs and predicts the crop and finally it sends backs the output to be displayed on the application.

Features of Android Application

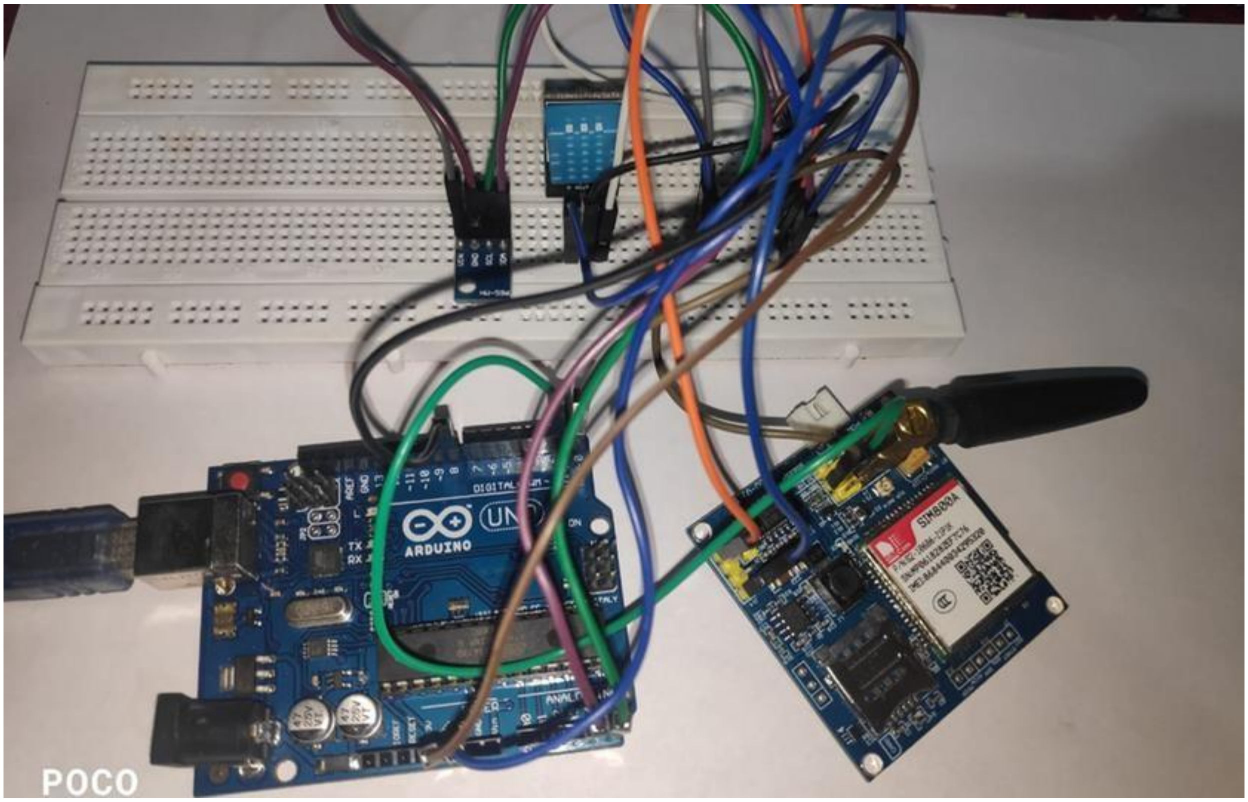
- ✓ Real time monitoring of weather components
- ✓ Graphic visualization of weather components
- ✓ Privacy of data using authenticated logins
- ✓ Prediction of suitable crops based on weather and soil conditions
- ✓ Tracking the lifecycle of the crop
- ✓ Knowing the best days suited for crop cultivation based on weather parameters.
- ✓ Easy to use
- ✓ Visually attractive.

The application also has a secondary feature of predicting the best days to sow seeds or harvest the crop according to the weather conditions of the region.

RESULTS AND DISCUSSION

HARDWARE SIMULATION

Hardware Setup



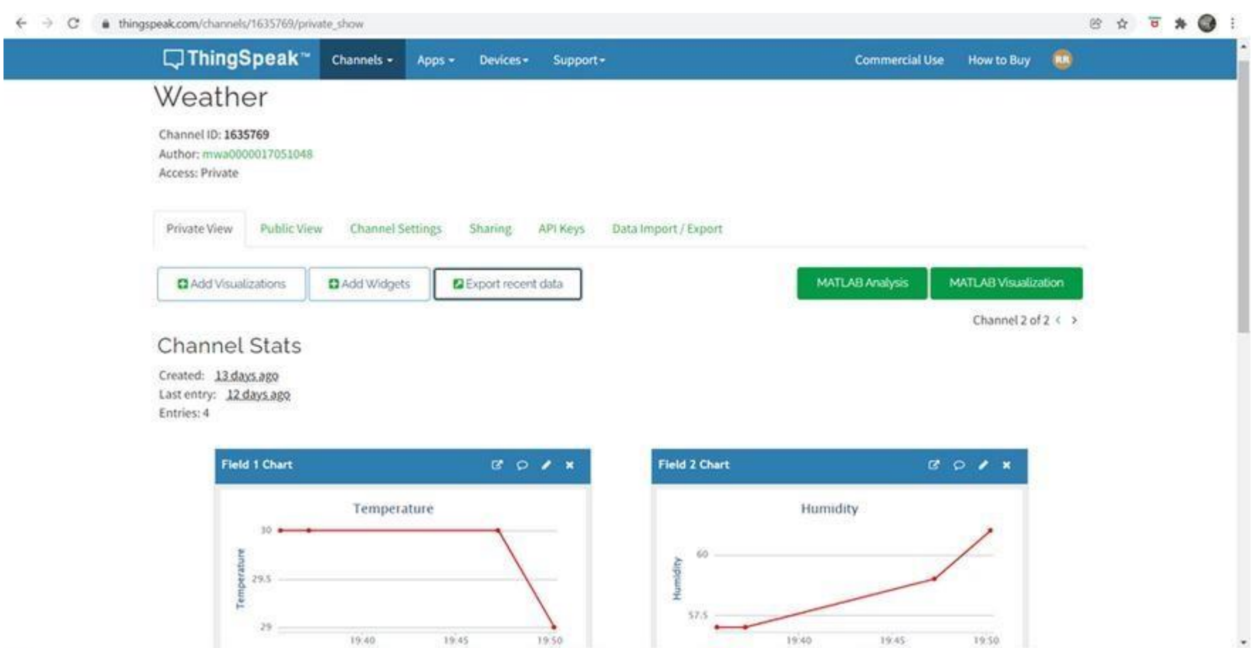
Connections

BMP-180 SENSOR	ARDUINO
Vin	5V
SCL	A5(Analog pin)
SDC	A4(Analog pin)
Gnd	Gnd

DHT11 SENSOR	ARDUINO
Vin	5V
DATA	D8 (digital pin)
Gnd	Gnd

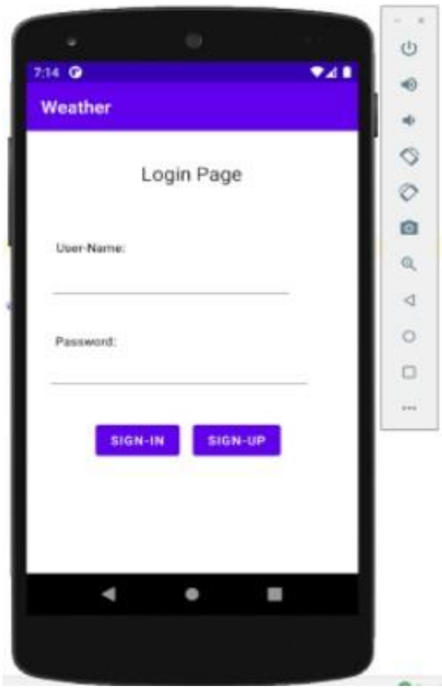
Arduino	GSM module
Digital pin RX	TX pin
Digital pin TX	RX pin
VCC – 5v	VCC
Ground	Ground

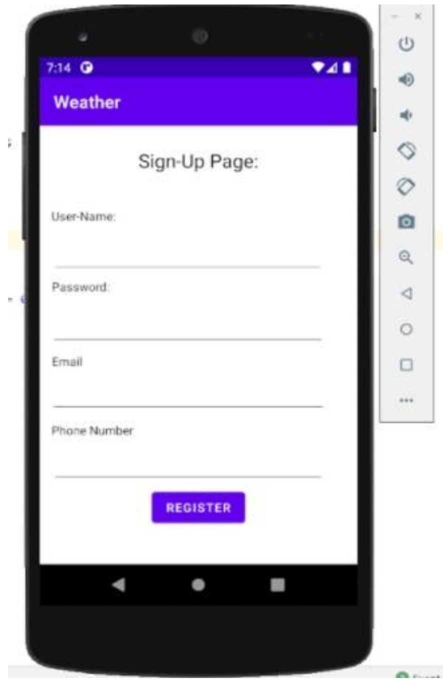
Cloud



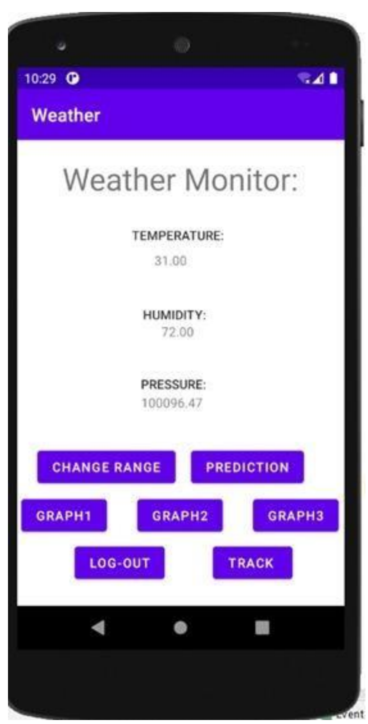
SOFTWARE SIMULATION

Login/Signup Page

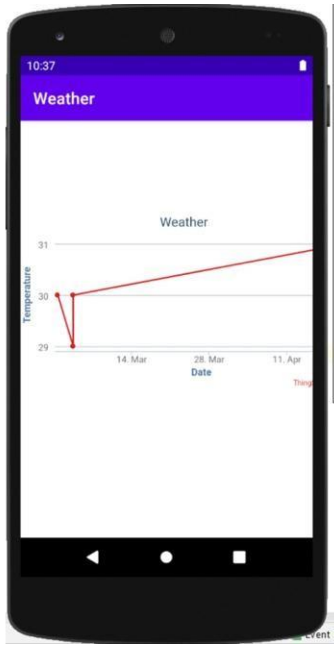




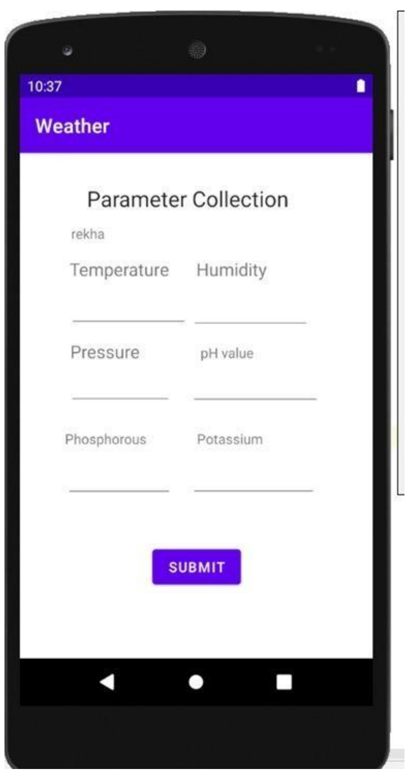
Realtime Weather Montioring

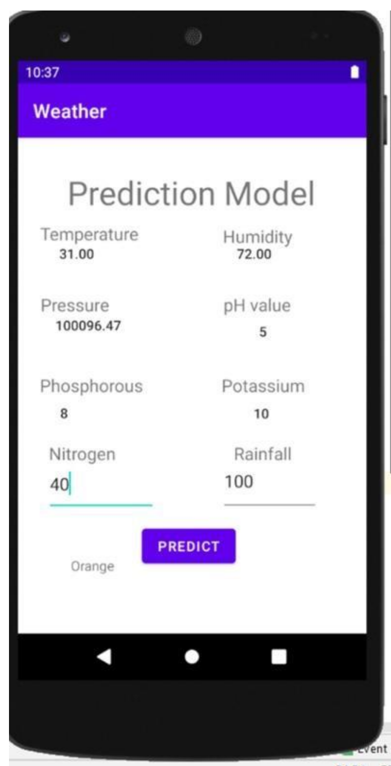


Weather Visualization

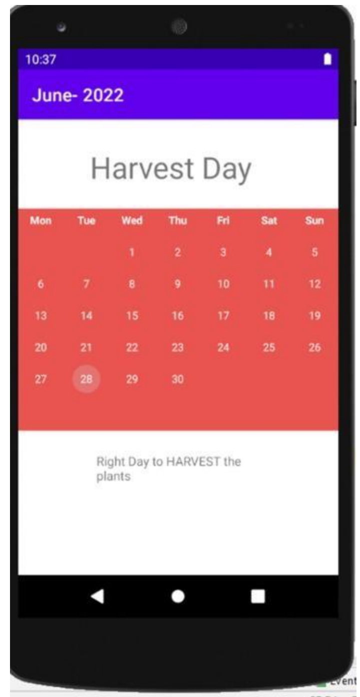


Prediction





Tracking the crop



CONCLUSION AND FUTURE ENHANCEMENTS

The project aims at improving yield of the crops and making better and healthy crops, giving profits to the farmer. With the help of IOT and machine learning the geographical conditions of a place can be studied and efficiently combined with agriculture. An additional feature of the system to track the progress of the crop makes the system unique and more user friendly. The future enhancements of the project are:

- ✓ More crops could be added as predictors
- ✓ Better machine learning approaches such as deep learning and convolutional neural network can be applied.
- ✓ The tracking feature could be universal for all the crops
- ✓ Extending the application to other operating systems like ios, etc

REFERENCES

[1] Umamaheswari S, Sreeram S, Kritika N, Prasanth DJ, "BIoT: Blockchain-based IoT for Agriculture", 11th International Conference on Advanced Computing (ICoAC), 2019 Dec 18 (pp. 324-327). IEEE.

[2] Jain A. "Analysis of growth and instability in the area, production, yield, and price of rice in India", Journal of Social Change and Development, 2018;2:46-66

[3] Manjula E, Djodiltachoumy S, "A model for prediction of crop yield" International Journal of Computational Intelligence and Informatics, 2017 Mar;6(4):2349-6363.

[4] Sagar BM, Cauvery NK., "Agriculture Data Analytics in Crop Yield Estimation: A Critical Review", Indonesian Journal of Electrical Engineering and Computer Science, 2018 Dec;12(3):1087-93.

[5] Wolfert S, Ge L, Verdouw C, Bogaardt MJ, "Big data in smart farming– a review. Agricultural Systems", 2017 May 1;153:69-80.

[6] Jones JW, Antle JM, Basso B, Boote KJ, Conant RT, Foster I, Godfray HC, Herrero M, Howitt RE, Janssen S, Keating BA, "Toward a new generation of agricultural system data, models, and knowledge products: State of agricultural systems science. Agricultural systems", 2017 Jul 1;155:269-88.

[7] Johnson LK, Bloom JD, Dunning RD, Gunter CC, Boyette MD, Creamer NG, "Farmer harvest decisions and vegetable loss in primary production. Agricultural Systems", 2019 Nov 1;176:102672.

[8] Kumar R, Singh MP, Kumar P, Singh JP, "Crop Selection Method to maximize crop yield rate using a machine learning technique", International conference on smart technologies and management for computing, communication, controls, energy, and materials (ICSTM), 2015 May 6 (pp. 138-145). IEEE.

[9] Sriram Rakshith.K, Dr.Deepak.G, Rajesh M, Sudharshan K S, Vasanth S, Harish Kumar N, "A Survey on Crop Prediction using Machine Learning Approach", In International Journal for Research in Applied Science & Engineering Technology (IJRASET), April 2019, pp(3231-3234)

[10] Veenadhari S, Misra B, Singh CD, "Machine learning approach for forecasting crop yield based on climatic parameters", In 2014 International Conference on Computer Communication and Informatics, 2014 Jan 3 (pp. 1-5). IEEE.
[11] Ghadge R, Kulkarni J, More P, Nene S, Priya RL, "Prediction of crop yield using machine learning", Int. Res. J. Eng. Technol. (IRJET), 2018 Feb;5.
[12] Priya P, Muthaiah U, Balamurugan M, "Predicting yield of the crop using machine learning algorithm", International Journal of Engineering Sciences & Research Technology, 2018 Apr;7(1):1-7.
[13] S. Pavani, Augusta Sophy Beulet P., "Heuristic Prediction of Crop Yield Using Machine Learning Technique", International Journal of Engineering and Advanced Technology (IJEAT), December 2019, pp (135-138)
[14] https://web.dev/progressive-web-apps/
[15] https://www.wikipedia.org/
[16] Plewis I, "Analyzing Indian farmer suicide rates", Proceedings of the National Academy of Sciences, 2018 Jan 9;115(2): E117.
[17] Nishant, Potnuru Sai, Pinapa Sai Venkat, Bollu Lakshmi Avinash, and B. Jabber. "Crop Yield Prediction based on Indian Agriculture using Machine Learning." In 2020 International Conference for Emerging Technology (INCET), pp. 1-4. IEEE, 2020.
[18] Kale, Shivani S., and Preeti S. Patil. "A Machine Learning Approach to Predict Crop Yield and Success Rate." In 2019 IEEE Pune Section International Conference (PuneCon), pp. 1-5. IEEE, 2019.
[19] Kumar, Y. Jeevan Nagendra, V. Spandana, V. S. Vaishnavi, K. Neha, and V. G. R. R. Devi. "Supervised Machine learning Approach for Crop Yield Prediction in Agriculture Sector." In 2020 5th International Conference on Communication and Electronics Systems (ICCES), pp. 736-741. IEEE, 2020.
[20] Nigam, Aruvansh, Saksham Garg, Archit Agrawal, and Parul Agrawal. "Crop yield prediction using machine learning algorithms." In 2019 Fifth International Conference on Image Information Processing (ICIIP), pp. 125-130. IEEE, 2019.
[21] Bang, Shivam, Rajat Bishnoi, Ankit Singh Chauhan, Akshay Kumar Dixit, and Indu Chawla. "Fuzzy logic based crop yield prediction using temperature and rainfall parameters predicted through ARMA, SARIMA, and ARMAX models." In 2019 Twelfth International Conference on Contemporary Computing (IC3), pp. 1-6. IEEE, 2019.

APPENDIX

Arduino code:

```
#include <dht.h>
#include <Wire.h>
#include <SFE_BMP180.h>

SFE_BMP180 bmp180;
dht DHT;

#define DHT11_PIN 7

void setup(){
  Serial.begin(9600);
    bool success = bmp180.begin();

  if (success) {
    Serial.println("BMP180 init success");
  }


}

void loop(){
  int chk = DHT.read11(DHT11_PIN);

  if(DHT.temperature!=-999)
  {
    Serial.print("Temperature = ");
    Serial.println(DHT.temperature);
  }

  if(DHT.humidity!=-999)
  {
    Serial.print("Humidity = ");
    Serial.println(DHT.humidity);
  }
  char status;
  double T, P;
  bool success = false;

  status = bmp180.startTemperature();

  if (status != 0) {
    delay(1000);
```

```
status = bmp180.getTemperature(T);
    if (status != 0) {
      status = bmp180.startPressure(3);

      if (status != 0) {
        delay(status);
        status = bmp180.getPressure(P, T);

        if (status != 0) {
          Serial.print("Pressure: ");
          Serial.print(P*100);
          Serial.println("Pa");


        }
      }
    }
  }

  Serial.println("AT");
  delay(1000);

  Serial.println("AT+CPIN?");
  delay(1000);

  Serial.println("AT+CREG?");
  delay(1000);

  Serial.println("AT+CGATT?");
  delay(1000);

  Serial.println("AT+CIPSHUT");
  delay(1000);

  Serial.println("AT+CIPSTATUS");
  delay(2000);

  Serial.println("AT+CIPMUX=0");
  delay(2000);

  ShowSerialData();

  Serial.println("AT+CSTT=\"airtelgprs.com\"");//start task and setting the APN,
  delay(1000);

  ShowSerialData();

  Serial.println("AT+CIICR");//bring up wireless connection
  delay(3000);
```

```
  ShowSerialData();

  Serial.println("AT+CIFSR");//get local IP adress
  delay(2000);

  ShowSerialData();

  Serial.println("AT+CIPSPRT=0");
  delay(3000);

  ShowSerialData();

  Serial.println("AT+CIPSTART=\"TCP\",\"api.thingspeak.com\",\"80\"");//start up the connection
  delay(6000);

  ShowSerialData();

  Serial.println("AT+CIPSEND");//begin send data to remote server
  delay(4000);
  ShowSerialData();

  String str="GET https://api.thingspeak.com/update?api_key=HITIG0DMLA7F9FLD&field1=" +
String(DHT.temperature) +"&field2="+String(DHT.humidity)+"&field3="+String(P*100);
  Serial.println(str);
  Serial.println(str);//begin send data to remote server

  delay(4000);
  ShowSerialData();

  Serial.println((char)26);//sending
  delay(5000);//waitting for reply, important! the time is base on the condition of internet
  Serial.println();

  ShowSerialData();

  Serial.println("AT+CIPSHUT");//close the connection
  delay(100);
  ShowSerialData();

  delay(1000);

}
void ShowSerialData()
{
  while(Serial.available()!=0)
  Serial.write(Serial.read());
  delay(5000);

}
```

Machine Learning Model:

```
import numpy as np
import pandas as pd
import pickle
import io
import sklearn
from google.colab import files
uploaded = files.upload()
data = pd.read_csv(io.BytesIO(uploaded['Crop_recommendation.csv']))
encoder = LabelEncoder()
data.label = encoder.fit_transform(data.label)
features = data.drop("label", axis=1)
target = data.label
X_train, X_test, y_train, y_test = train_test_split(features, target, random_state=42)
rf = RandomForestClassifier(n_estimators=10, max_features=3, random_state=0).fit(X_train, y_train)
rf_pred= rf.score(X_test, y_test)
import pickle
pickle.dump(knn,open('model1.pkl','wb'))
```

Building Flask API:

```
from flask import Flask,request,jsonify
import pickle
import numpy as np

model=pickle.load(open('model1.pkl','rb'))

app=Flask(__name__)

@app.route('/')
def home():
    return "Hello World"
@app.route('/predict',methods=['POST'])
def predict():
    n=request.form.get('n')
    p=request.form.get('p')
    k=request.form.get('k')
    t=request.form.get('t')
    h=request.form.get('h')
    ph=request.form.get('ph')
    r=request.form.get('r')

    input_query=np.array([[n,p,k,t,h,ph,r]])
    result=model.predict(input_query)[0]
    return jsonify({'placement':str(result)})

if __name__ =='__main__':
    app.run(debug=True)
```

Android application:

```
package com.example.weather;

import android.content.Intent;
import android.nfc.Tag;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONException;
import org.json.JSONObject;

import java.util.HashMap;
import java.util.Map;

public class Main5Activity extends AppCompatActivity {

    Button b;

    TextView temp,hum,press,ph_value,phos,potassium_value,result;
    EditText nitrogen, rain;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main5);
        b=(Button) findViewById(R.id.predict);
        temp = (TextView) findViewById(R.id.t);
        hum = (TextView) findViewById(R.id.h);
        press = (TextView) findViewById(R.id.textView28);
        ph_value = (TextView) findViewById(R.id.ph);
        phos = (TextView) findViewById(R.id.p);
        potassium_value = (TextView) findViewById(R.id.k);
        result=(TextView) findViewById(R.id.result);

        nitrogen = (EditText) findViewById(R.id.n);
        rain = (EditText) findViewById(R.id.r);
```

```
        Intent i=getIntent();
        String temperature=i.getExtras().getString("temperature1");
        String humidity=i.getExtras().getString("humidity1");
        String pressure=i.getExtras().getString("pressure1");
        String ph=i.getExtras().getString("ph1");
        String phosphorous=i.getExtras().getString("phosphorous1");
        String potassium=i.getExtras().getString("potassium1");

//        String rain="100";
//        String nitrogen="40";

        temp.setText(temperature);
        hum.setText(humidity);
        press.setText(pressure);
        ph_value.setText(ph);
        phos.setText(phosphorous);
        potassium_value.setText(potassium);

        String url = "https://tarp-1.herokuapp.com/predict";

        b.setOnClickListener(new View.OnClickListener() {


            @Override
            public void onClick(View view) {



                //hit the API -> Volley
                StringRequest stringRequest = new StringRequest(Request.Method.POST, url,
                        new Response.Listener<String>() {
                            @Override
                            public void onResponse(String response) {
                                try {
                                    JSONObject jsonObject = new JSONObject(response);
                                    String data = jsonObject.getString("placement");
                                    if(data.equals("0"))
                                    {
                                        result.setText("Apple");
                                    }
                                    else if(data.equals("1")){
                                        result.setText("Banana");
                                    }
                                    else if (data.equals("2")){
                                        result.setText("Black Gram");
                                    }
                                    else if (data.equals("3")){
                                        result.setText("Chickpeas");
                                    }
```

```
else if (data.equals("4")){
    result.setText("Coconut");
}
else if (data.equals("5")){
    result.setText("Coffee");
}
else if (data.equals("6")){

    result.setText("Cotten");
}
else if (data.equals("7")){

    result.setText("grapes");

}
else if (data.equals("8")){
    result.setText("Jute");
}
else if (data.equals("9")){
    result.setText("Kindey Beans");

}
else if (data.equals("10")){

    result.setText("Lentil");

}
else if (data.equals("11")){

    result.setText("Maize");
}
else if (data.equals("12")){

    result.setText("Mango");

}
else if (data.equals("13")){
    result.setText("Moth Beans");
}
else if (data.equals("14")){
    result.setText("Moung Beans");
}
else if (data.equals("15")){

    result.setText("Musk melon");

}
```

```
                else if (data.equals("16")){
                    result.setText("Orange");

                }


                else if (data.equals("17")){
                    result.setText("Papaya");
                }
                else if (data.equals("18")){

                    result.setText("Pigeon peas");

                }
                else if (data.equals("19")){

                    result.setText("Pomegranate");

                }
                else if (data.equals("20")){
                    result.setText("Rice");

                }
                else if (data.equals("21")){

                    result.setText("Watermelon");

                }

                else{
                    result.setText("No crop found");
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    },
    new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Toast.makeText(Main5Activity.this, error.getMessage(),
Toast.LENGTH_SHORT).show();
        }
    }){
```

```
            @Override
            protected Map<String,String> getParams(){

                Map<String,String> params = new HashMap<String,String>();
                params.put("t",temp.getText().toString());
                params.put("h",hum.getText().toString());
                //params.put("p",press.getText().toString());
                params.put("ph",ph_value.getText().toString());
                params.put("p",phos.getText().toString());
                params.put("k",potassium_value.getText().toString());
                params.put("r",rain.getText().toString());


                params.put("n",nitrogen.getText().toString());

                Log.i("temperature",temp.getText().toString());
                Log.i("humidity",hum.getText().toString());
                Log.i("ph",ph_value.getText().toString());
//                Log.i("param",new Gson().toJson(params));


                return params;
            }
        };
        RequestQueue queue = Volley.newRequestQueue(Main5Activity.this);
        queue.add(stringRequest);




    }
  });


   }
}
```