PROJECT REPORT

Entitled

"IoT based plant and soil monitoring system"

Submitted to the Department of Electronics Engineering
In Partial Fulfillment of the Requirement for the Degree of

Bachelor of Technology
(ELECTRONICS & COMMUNICATION)

: Presented & Submitted By:

Mr. Deep Jariwala (Roll No. U16EC027)
Mr. Rajat Kumar Panigrahi (Roll No. U16EC057)
Mr. Monu Chouhan (Roll No. U16EC061)
Mr. Kaustubh Shah (Roll No. U16EC140)
B. TECH. IV (EC), 8th Semester

: Guided By:

Prof. N. B. Kanirkar
Associate Professor, ECED.

(Year: 2019-20)

DEPARTMENT OF ELECTRONICS ENGINEERING
Sardar Vallabhbhai National Institute of Technology

Sardar Vallabhbhai National Institute of Technology

Surat-395 007, Gujarat, INDIA.

ELECTRONICS ENGINEERING DEPARTMENT

CERTIFICATE

This is to certify that the **PROJECT REPORT** entitled **"IoT based plant and soil monitoring system"** is presented & submitted by Candidate **Mr. Deep Jariwala, Mr. Rajat Kumar Panigrahi, Mr. Monu Chouhan, Mr. Kaustubh Shah** bearing **Roll No. U16EC027, U16EC057, U16EC061, U16EC140 of B.Tech. IV, 8th Semester** in the partial fulfillment of the requirement for the award of **B. Tech.** degree in **Electronics & Communication Engineering** for academic year 2019-20.

They have successfully and satisfactorily completed their **Project Exam** in all respect. We, certify that the work is comprehensive, complete and fit for evaluation.

Prof. N. B. Kanirkar
Associate Professor &
Project Guide

PROJECT EXAMINERS:

Name of Examiner	Signature with date
1. Dr. U.D. Dalal	____________________
2. Dr. S. Gupta	____________________
3. Dr. S. N. Shah	____________________

Dr. A. D. Darji
Associate Professor
Head, ECED, SVNIT

DEPARTMENT SEAL
(June - 2020)

ACKNOWLEDGEMENT

First and foremost, we would like to express our deepest gratitude to our project preliminary guide, **Prof. N. B. Kanirkar,** for his support and constant guidance during our research period of this project. His knowledge and experience in this particular field was very helpful in obtaining the correct path and valuable resources required for this research. He has given us the freedom to work and explore IoT through many research papers and journals. We are extremely thankful to our project guide for giving us time for reporting and preparing presentations. We would also express our deepest gratitude to **Dr.Z.M. Patel** and **Dr.A.D. Darji** for providing us the platform for research and presenting our work as a project preliminary examination. Finally, we would thank all the teaching and non-teaching staff of electronics and communication engineering department for providing cooperation in every possible manner.

DEEP JARIWALA - U16EC027

RAJAT KUMAR PANIGRAHI - U16EC057

MONU CHOUHAN - U16EC061

KAUSTUBH SHAH - U16EC140

ABSTRACT

TITLE: IOT BASED PLANT AND SOIL MONITORING SYSTEM

Internet of Things (IoT) is one of the most demanding fields for building a large-scale project based on data collection. Moreover, by incorporating data science and machine learning techniques with this domain, we can create a system that could easily accomplish any application-based project. Agriculture is one of the areas where these techniques can be used to achieve great results. However, this sector needs an amalgamation of the latest technology to provide better results. We have built a system that provides overall coverage of weather and soil conditions using these domains.

The proposed system is an intelligent wireless system that inculcates the general topology of data acquisition and processing system in the form of a project titled "IoT based plant and soil monitoring system". We have created a NodeMCU based hardware system that is connected to various environmental sensors to fetch data about different physical parameters of the farm, such as temperature, humidity, and soil moisture content. NodeMCU is connected to the Internet via Wi-Fi. A hardware unit consists of all the sensors along with NodeMCU. These units are placed at various locations throughout the farm. Hence, the entire farm can be monitored, and better results can be obtained.

The system uses Firebase Cloud Storage as the Database. The data collected from the sensors is stored in the Firebase Real-Time database over Wi-Fi. We have developed a mobile application that will display all the data to the user in real-time. It is a user-friendly application that can be accessed over the Internet. The mobile app offers several features such as sharing the data in PDF file format, receiving notifications when some parameter of the farm goes off-limits and showing the variation of these parameters in graphical form. The application also supports local languages like Gujarati and Hindi.

The next part, which is incorporated into our system, is the machine learning algorithms that work on data set to predict whether it will rain or not. Furthermore, using linear regression and data science techniques, we have built a model to show the correlation of various climate parameters and predicted the amount of crop production based on rainfall

and land area. The results have been demonstrated with accuracy and different visualization techniques based on data science that can help the user to differentiate between actual and predicted values.

The report also shows a brief comparison of various types of sensors. We have opted for the most efficient and reliable components that form the required system from which data is collected and stored on the cloud. The report involves a literature survey of various other projects and IEEE papers to compare our system's reliability. Moreover, it consists of a set of raw data from the environmental sensors and screenshots of the application that display the data and visualization of the dataset to encounter a pattern that could be effective in choosing the machine learning algorithm.

Student Name: **Mr. DEEP JARIWALA**
Mr. RAJAT PANIGRAHI
Mr. MONU CHAUHAN
MR. KAUSTUBH SHAH
Roll No. **U16EC027**
U16EC057
U16EC061
U16EC140
Guide Name: **Prof. N.B. Kanirkar**
Date of Exam: 17-06-2020
Timeslot of Project Exam: 10.00 am to 1.00 pm
Report Submission Date: 05-06-2020

Examiner Name: **Dr. U. D. Dalal**
Examiner Name: **Dr. S. Gupta**
Examiner Name: **Dr. S. N. Shah**

*****

INDEX

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1

INTRODUCTION

1.1 Overview:

As time is going on, technology is improving at a faster and faster rate. With the advancement of technology in computers and the internet, our lives have become more comfortable and simpler. With one-button, we can order food by sitting at home, shop online, do banking transactions, book a flight or a train or bus or a cab, book a hotel, even control vehicles, aircraft, missiles, and many other things. So, the demand for this simplicity and ease of life is increasing. The internet of things (IoT) is one idea to make life easier as it refers to the communication between devices collecting different information from the real world. The communication using IoT can be done by sensors and giving internet access to the tools. The real-world data collected by the devices is accessed over a server (cloud) via the internet. As the data is uploaded on the internet, it can be viewed from anywhere and anytime. Here, some operations can be done on the data, which will give information on whether there is any problem. If there is some problem, accordingly, action can be taken. For example, an automobile that has built-in sensors to alert the driver when tire pressure is low or suppose somebody has broken into your house. You are not there; then, in this case, not only would your alarm start to ring, but a message may also be sent to you stating that warning has gone off, etc. These are simple examples to demonstrate how IoT can be useful.

This project is based on monitoring plant conditions using IoT. In this project, sensors have been used to collect data, and this data is transferred to a firebase (server). This information can be accessed anywhere, using a mobile application. This project has been taken up with a thought of being able to implement this kind of technology in the agricultural field. Machine learning has been included in the project to help predict rainfall. The prediction of rainfall would help in increasing the productivity and selection of crops. The project aims to help ease the farmer’s burden, and loss as the farmers can access the information on the soil's condition, temperature and humidity, and rainfall from anywhere using their phones.

1.2 IoT

The concept of IoT started to gain some popularity in the summer of 2010. Information leaked that Google's Street View service had not only made 360-degree pictures but had also stored tons of data of people's WiFi networks. The same year, the Chinese government announced it would make the Internet of Things a strategic priority in their Five-Year-Plan. In October of 2013, IDC published a report stating that the Internet of Things would be an $8.9 trillion market in 2020. The term Internet of Things reached mass-market awareness when, in January 2014, Google announced to buy Nest for $3.2bn. Now, IoT is the hottest prospect in describing the new interconnected world. IoT is essentially a platform where embedded devices are connected to the internet so that they can collect and exchange data with each other. There are many advantages, disadvantages, and applications of IoT.

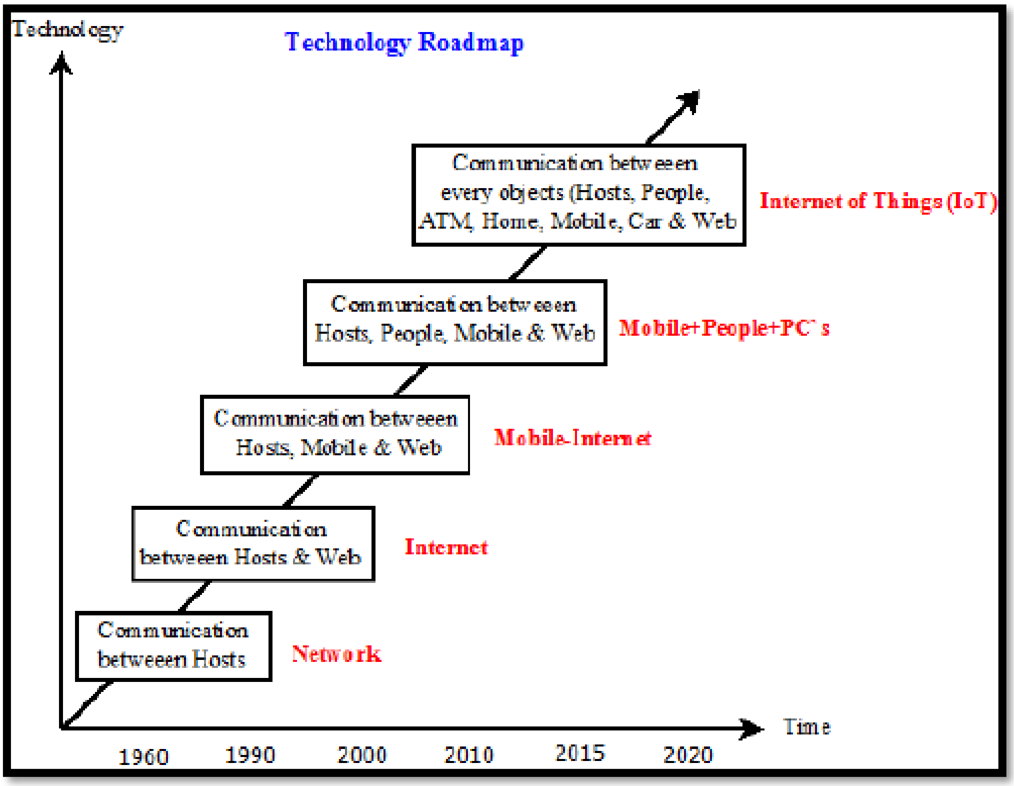


Fig 1.2(a) Roadmap of technology and development of IoT

1.2.1 Advantages:

Some of the advantages of IoT include:

• Information can be accessed from anywhere and anytime. Also, the devices can be monitored and controlled over the internet without any human effort.

• The communication between the devices would be improved.

• Transferring data packets over a connected network saves time and money;

• Automation enhances the quality of the tools, and there is no need for human intervention, meaning more profit for organizations.

1.2.2 Disadvantages:

Some disadvantages of IoT include:

• As the amount of information shared increases, the risk of a security breach increases. If there is any breach of security, a lot of damage can take place.

• For big organizations, the amount of IoT devices required is enormous.

• If there's a bug in the system, it's likely that every connected device will become corrupted.

Since there's no international standard of compatibility for IoT, it's difficult for devices from different manufacturers to communicate.

1.2.3 Application:

There are and can be many applications of IoT and are presently being used in various domains.

1. IoT smartwatches are being developed, which can be used to calculate calories burnt, heart rate, listen to songs, control calls, etc., all these can be seen on PC if we want, and their settings can be controlled from PC.

2. Home applications can be connected and controlled by our mobile phones. For example, you can control TV remote from portable, microwave oven from the mobile phone, suppose you have gone for vacation but have forgotten to switch off some light or fan or any other equipment, all can be monitored and controlled.

3. IoT based traffic management systems are being developed to find where more traffic is occurring, whether the rules are being followed properly or not etc.

4. Smart farming and automated irrigation systems are being developed to reduce human labor, increase crop yield, improve plant health, and understand soil. Many sensors are installed, used to measure the plant and soil reports, and are wirelessly sent to an IoT cloud.

5. IoT can be used in the field of medicine. For example, patients' heart rate can be monitored by sensors and sent to the doctor periodically for diagnosis. If there is any problem, then it will be notified, and the doctor would come to know about it, and hence, further action can be taken.

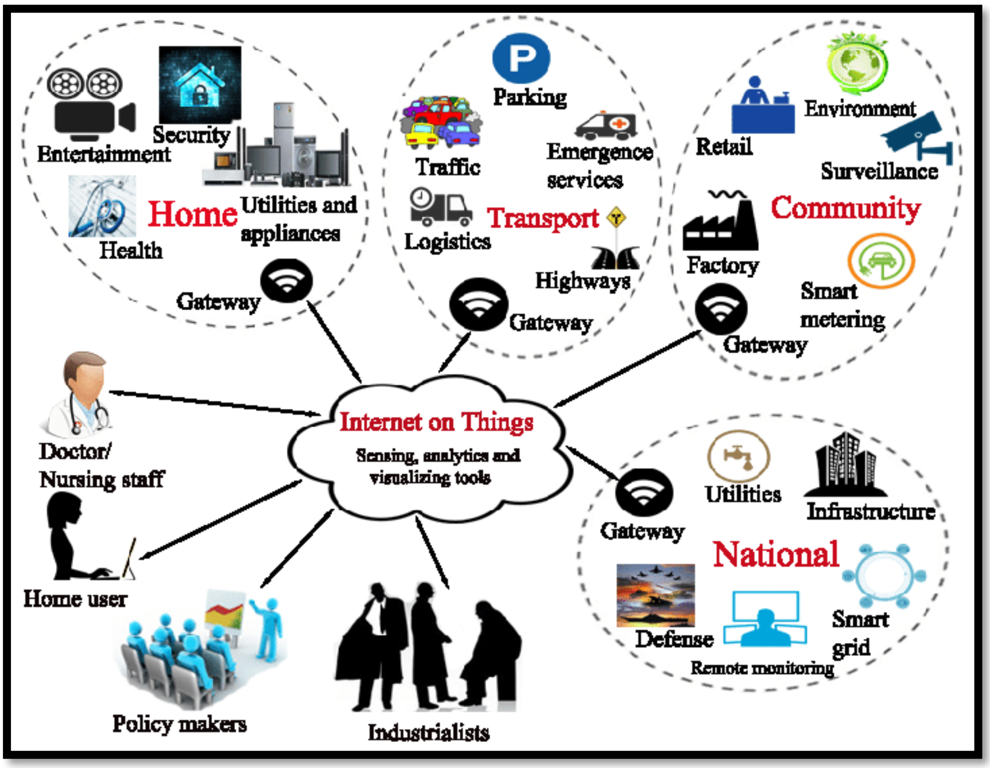


Fig1.2.3(a) Applications of IoT

1.3 Machine learning and data science

Machine learning is the application of artificial intelligence. A system can automatically learn and improve from experience. The methods which use machine learning are programmed in such a manner that they can remember past experiences and accordingly predict future results. This technology is attracting a lot of attention for its new approach and vast scope of improvement in the future. There are many benefits and applications of machine learning.

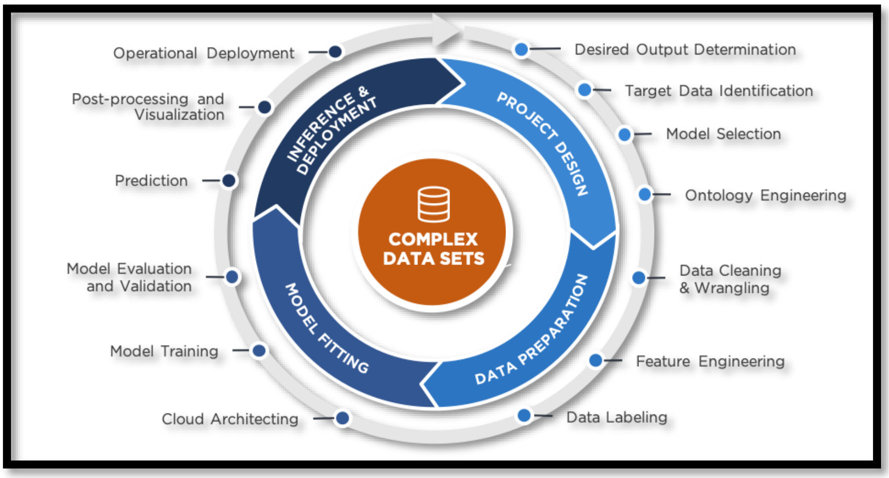


Fig 1.3(a) Machine Learning Lifecycle

In this project, machine learning has been used to predict rainfall. This prediction is useful for farmers as it would give them an idea as to when it might rain, and based on the amount of moisture, crops can be selected for maximum growth. In this way, the farmers would get enhanced productivity, and losses would be reduced. Two different algorithms are used in this project. Linear Regression is used to find how much production can be done in the farm area, and the other is Logistic Regression, which is used to predict rainfall.

Machine learning uses data science, which is to visualize and analyze data. The processes in machine learning are that the first data is collected, then cleaned and processed (this is where data sciences are used). Then, tests are conducted based on some test data, and thus, the prediction is made. We are now checking starts to find the difference or error in the received results and the actual results. Finally, correction is implemented, and results are concluded.

Machine learning can be categorized as supervised and unsupervised learning. Supervised learning is when a system is given some desired value, and it should accommodate changes in itself automatically to get the desired results. Unsupervised learning is to leave it to the system to find the best-suited result. Using algorithms, this can be implemented. There are many advantages of machine learning; some of them are

1. The ability of systems to improve over time
2. No human involvement
3. Quickly identifies trends and patterns.
4. Machine learning algorithms are good at handling data that are multidimensional and multi-variety.

There are many applications of machine learning. Some of them are:

1. Image recognition
2. Prediction of rainfall
3. Speech recognition
4. Virtual personal assistant
5. Email spam and malware filtering
6. Online customer support
7. Search engine result refining

Unlike data in the traditional systems, which was mostly structured, today, most of the data is unstructured or semi-structured. This data is generated from different sources like financial logs, text files, multimedia forms, sensors, and instruments. The data accumulated is huge and is varied. So, more complex and advanced analytical tools and algorithms for processing are required. Thus, Data science is a "concept to unify statistics, data analysis, machine learning, and their related methods" in order to "understand and analyze actual phenomena" with data. It has many applications in today's life. They are:

1. <u>Fraud and risk detection</u>: Over the years, banking companies learned to divide and conquer data via customer profiling, past expenditures, and other essential

variables to analyze the probabilities of risk and default. Moreover, it also helped them to push their banking products based on the customer's purchasing power.

2. Healthcare: Procedures such as detecting tumours, artery stenosis, organ delineation employs machine learning methods, support vector machines (SVM), content-based medical image indexing, and wavelet analysis for solid texture classification. Data science is used for drug development and also for virtual assistants for patients.
3. Internet search: All the search engines (including Google) make use of data science algorithms to deliver the best result for our searched query in a fraction of seconds.
4. Advanced Image Recognition: This automatic tag suggestion feature (on Facebook, for example) uses a face recognition algorithm.

1.4 Motivation:

IoT exploits recent advances in software, falling hardware prices, and modern attitudes towards technology. Its new and advanced elements bring significant changes in the delivery of products, goods, and services; and the social, economic, and political impact of those changes.

IoT can be used in agriculture for improving crop yield, maintaining crop health, soil moisture. By implementing IoT, we can access the farm's data anywhere anytime, and we can also control it depending on requirements. The scope for improvement in this field is vast; it would only make life simpler, more comfortable with more benefits.

This project involves hardware as well as software knowledge with the new technological concept, and it can be implemented in our day to day life.

****

CHAPTER-2

LITERATURE SURVEY

2.1 Overview:

This chapter focuses on the work and system developed in different research papers. IoT based plant and soil system is the primary keyword through which we have given the basic summary of a few research papers. The later part of the chapter has research papers based on weather and crop yield prediction using machine learning algorithms.

2.2 IoT based crop- field monitoring and irrigation automation:

An IoT system consists of internet-accessible devices that use embedded processors, sensors, and communication hardware. This hardware is used to collect, send, and act on data they get from their environments. IoT devices share the sensor data they collect by connecting to an IoT gateway or another edge device where data is either sent to the cloud to be analyzed or analyzed locally. Sometimes, these devices communicate with other related accessories and act on the information they get from one another. The tools do most of the work without human intervention, although people can interact with the devices -- for instance, to set them up, give them instructions or access the data.

- To interface the sensor with the system, Arduino UNO has been used, and data transferred to the NRF system.
- To monitor the field data and send it to the web servers NRF24LO1 transmitter and receiver and Ethernet connection at receiver ends are used.
- This system has two modules first for acquiring the data and second (i.e., NRF) for sending the data through the wireless network and at the receiver end another NRF module for receiving the data.[3]

2.3 Automatic plant escalation monitoring system using IoT:

The proposed system includes four layers, namely the monitoring layer, network layer, application layer, and UI layer. The monitoring layer helps to collect and aggregates the monitored information like temperature, humidity, pressure soil moisture, and plant images through the sensor devices. And also, a classifier is built using a convolutional neural network (CNN) to classify whether a plant is infected or not for the image captured by a raspberry pi. The raspberry pi camera module is fixed in the camera port of the raspberry pi to obtain the current model of the plant, which will help to detect whether the plant is affected by diseases or not. The role of the network layer is to transmit the aggregated data to the cloud server through a communication medium. The application layer maintains the historical information about the plant and is compared with the currently received aggregated information from the agriculture field to check the status of the plant. Finally, the UI layer provides the necessary user interaction, via a mobile app with the warning message. The UI layer provides the user interaction from which the user can able to get a warning message when his plant gets affected or lost its moisture content, humidity, temperature, and pressure of the soil. [6]

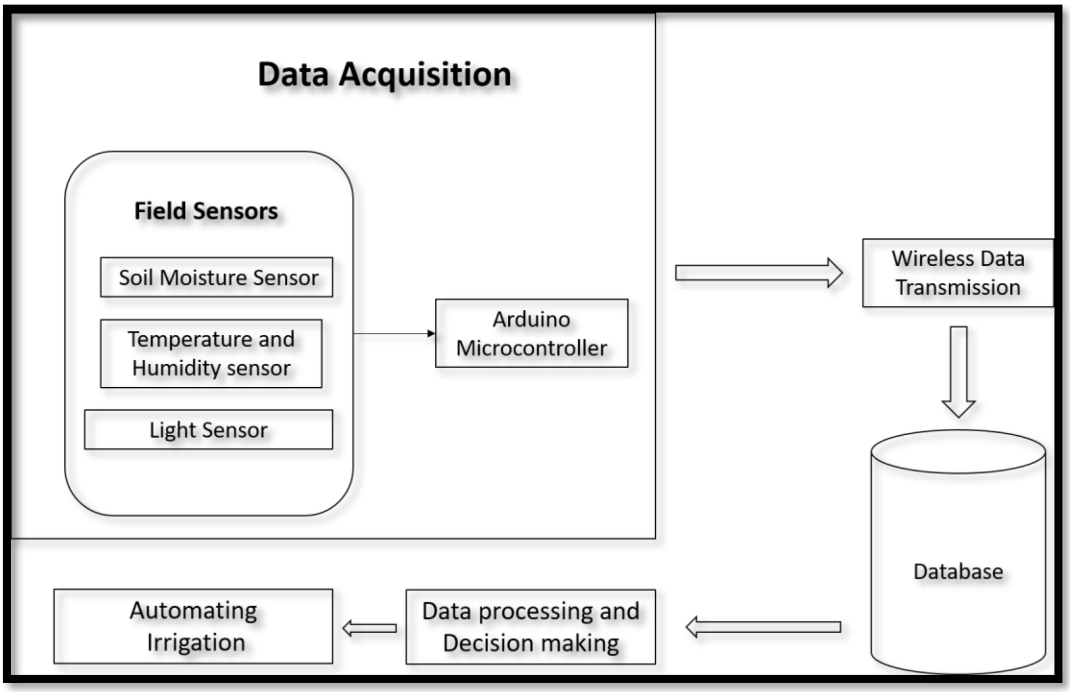


Fig 2.3(a) Block diagram [5]

The agricultural field is continuously monitored by the surveillance camera called raspberry pi camera module v-2 for better monitoring of plant in both day and night. Initially, the sensors are used to collect the field information such as temperature, soil moisture, humidity, and pressure of the soil in the agricultural field. The data sensed from the individual sensors are aggregated periodically based on their relevant parameter specific types. This aggregated information is stored in the cloud via raspberry pi 3. The historical information about the growth of the plant, such as soil temperature, soil moisture, humidity, and pressure value is maintained in the cloud server. Similarly, the information about the plant features under different environmental conditions is also stored in the cloud server. These complete monitoring activities are analyzed at the cloud server, and an appropriate suggestion message will be provided to the farmers through their mobile using the android app and thus increasing the yield.

2.4 Smart irrigation system-based Thing speak and Arduino:

The same implementation with other software and hardware can be stated as:

- Arduino UNO: is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins and six analog inputs; it connects to a computer via a USB cable or power it with an AC / DC adapter or battery.
- Thing Speak platform: This is an IoT analytics platform service that can you view and analyze live data in the cloud and also give you the ability to execute MATLAB code.
- Wi-Fi module ESP8266: is a self-contained SOC with an integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network. The Wi-Fi serial module works in both directions: it uses a TX / RX serial link to receive and send data.

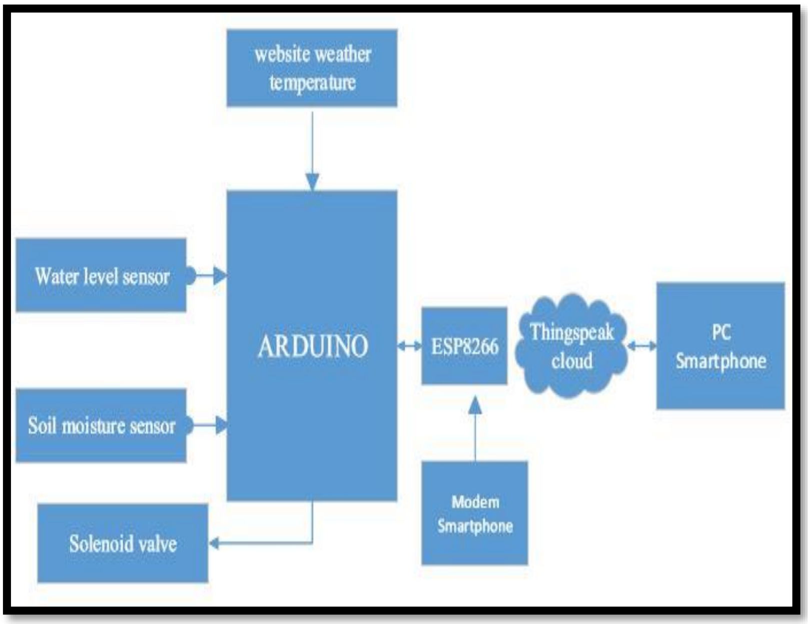


Fig 2.4(a) Block diagram [4]

2.5 IoT based smart Irrigation monitoring and controlling system:

The proposed system helps the user to improve the quality and quantity of their farm yield by sensing ambient temperature and humidity values, soil moisture from the field without any human intervention. By using the concept of the IoT system can be more efficient. The system contains wireless sensor units node1 node2 placed in the field to acquire the real-time values, a master node to receive and transmit acquired information to the control section, and a control section that controls the drips for the watering subsystem. Each node includes temperature, humidity, soil moisture, and microcontroller, and relay switching unit. The sensed data from each node is transmitted to the master node via ZigBee. The received data from the master is stored at the cloud server. The cloud server performs decision making by comparing sensed values and predefined threshold values as per crop selection. Once data is processed, and the decision is determined at the control section and takes action accordingly. [4]

2.6 Rainfall prediction using machine learning:

Artificial intelligence has been extensively used in all the applications, and weather forecasting is not an exception. When it comes to weather forecasting, rainfall prediction is one of the most commonly used research areas as numerous lives, and property damages occur due to this. Previous rainfall prediction models that are widely used make use of many complicated mathematical instruments that were insufficient to get a higher classification rate. So new methods for predicting monthly rainfall using linear regression analysis. By collecting quantitative data about the current state of the atmosphere, rainfall predictions can be a possibility. Numerous machine learning algorithms can learn complex mappings from inputs to outputs, based solely on samples and require limited. The rainfall prediction method utilizes the variation in the conditions in past years. By using the regression model considering various parameters such as temperature, humidity, and wind prediction of the amount of rainfall is possible.

The proposed model tends to forecast rainfall based on the previous records of a particular geographic area; therefore, this prediction will prove to be much reliable. The performance of the model is more accurate when compared with traditional rainfall prediction systems. The proposed method makes use of Linear Regression along with the use of a dataset of different parameters. It acts as the input, and the toolbox reads the file, which is converted into separate data text where the 1st year's plot of rainfall

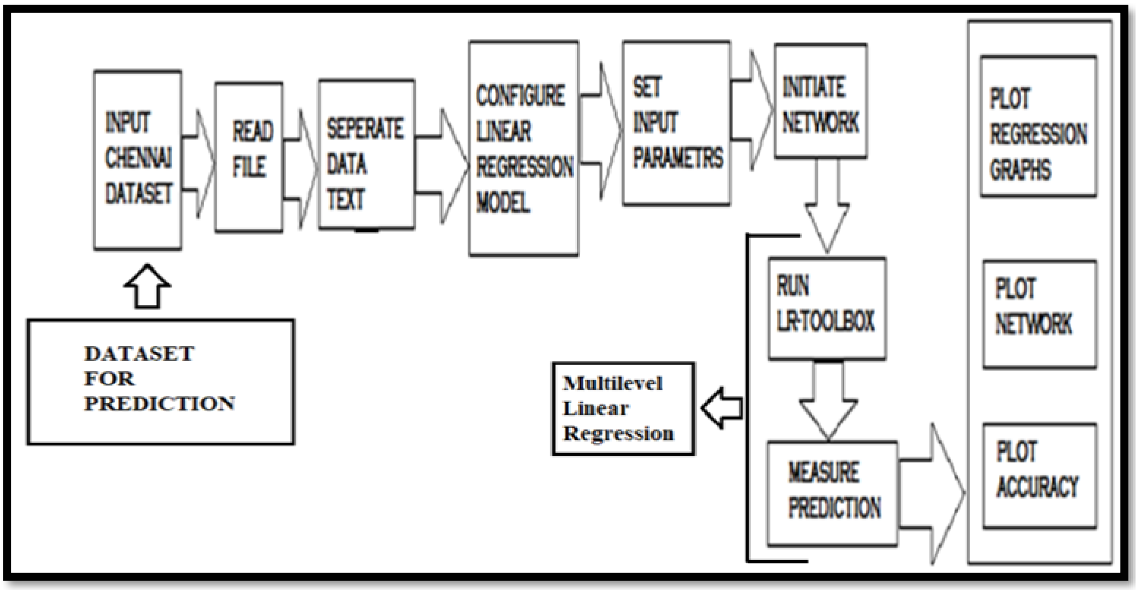


Fig. 2.6(a) Block diagram of the proposed model [9]

The dataset is trained by using Linear Regression, where various parameters are given to initiate the entire network. The prediction values are measured using the toolbox, and the graphs are plotted for the obtained predicted values. When it comes to the Linear Regression Model, the rainfall dataset is used as the input where the pre-processing stage happens. The feature is extracted by making use of the Linear Regression Model.

The values or variable set that frequently occurs in the model is identified and separated for getting the list of the frequent items. Various parameters are obtained while using these models, such as the temperature, the humidity, and the rainfall of a given geographic area. When it comes to Linear Regression model, the rainfall dataset is used as the input where the pre-processing stage happens. The feature is extracted by making use of the Linear Regression Model.

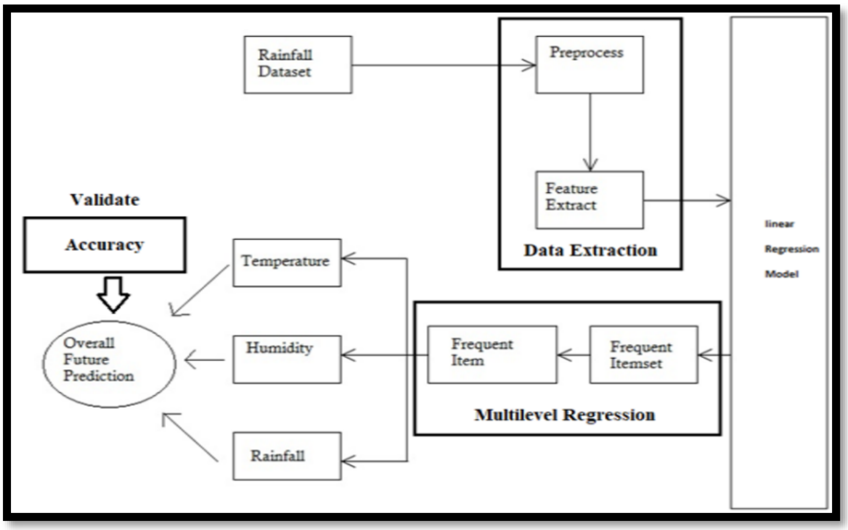


Fig. 2.6(b) Architecture of the proposed model. [9]

2.7 Weather prediction using linear regression algorithm:

The weather prediction parameters like humidity, temperature, and Moisture, are collected from the various stations of the meteorological department. The next step involved in the model is the selection of the data, and processing of the selected data is made by the machine learning algorithm to predict the weather

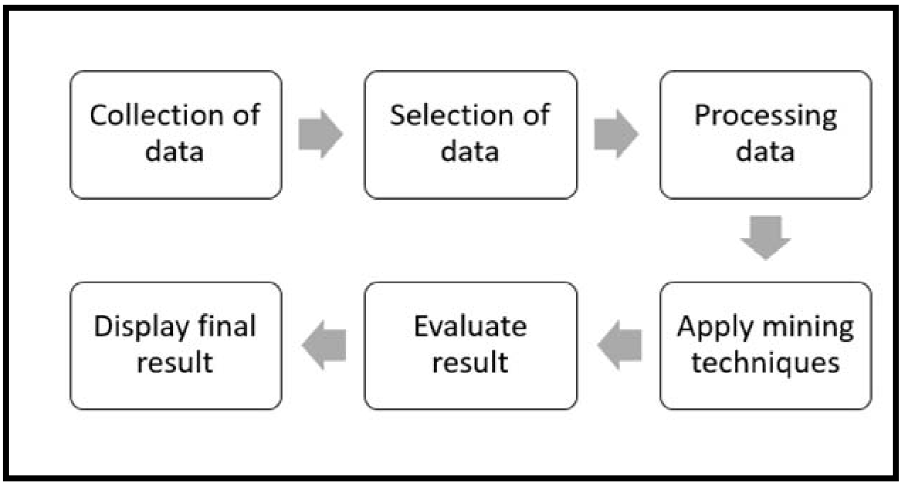


Fig. 2.7(a) System block diagram. [11]

In this paper, three types of weather parameters: humidity, Dew-point, and temperature, are determined using the prediction algorithm. The optimization algorithms used are gradient descent and the ordinary equation. The normal equation is an error minimization technique used in the linear regression problem with a least-squares cost function where the parametric and the total error is calculated. There is minimal deviation in the output from the actual value by using the standard equation algorithm. The error produced by the gradient descent method is very significant compared to the error produced by the standard equation method.

This section demonstrates the Rainfall Prediction Model's most prominent features and explains how they can be applied to the input dataset to get the results.

1) Hardware Requirements: The minimum hardware requirements of Rainfall prediction are a 500-Megahertz CPU and 128megabytes of RAM. Also, because this model uses python libraries to speed up the forecast. There should be considerable memory requirements for training and testing of data.

2) Software Requirements: Windows, macOS or Linux operating system
Linear regression is used to predict the rainfall. For that data required for the system to predict rainfall is previous year rainfall data, average temperature, and humidity over the particular area. This collected data is used to formulate the equation for predicting the rain by calculating the average temperature and humidity in that area. The modified version of Linear Regression is used to perform the prediction of rainfall in our system.
The latest updated coefficients to forecast the test data, and this produces accurate forecast values.

A. Server:

- Obtain information
- Display information on the user's app
- Update information according to user instructions

1) Obtain Sensor Information

The server takes information about previous rainfall.

2) Display Sensor Information on User's App

The server sends the data to the user and gives the user all the information.

3) Update information according to user instructions

The server's job is to use data from a database as per instructions sent by the user through the website.

B. User:

- Registration
- Login
- Access to the server

1) Registration

Users will register to the system with standard information.

2) Login

For login, the user will enter the user name and password; if entered information is correct, the system will redirect it to the home page; otherwise, it will show an error message.

3) Access to the server

After login, the user will see the detail about temperature

Checks for the needed information.

2.8 Crop yield prediction using machine learning algorithm:

In ML, the computer learns automatically from data & information using a different computer algorithm. A computer doesn't need to explicitly programmed. These can be improved & change algorithms by themselves. The process of working with machine learning is the gathering of data, Data preparation, choose a model, Training, Evaluation, Parameter training, and Prediction.

Proposed system:

Dataset Used

Data sets used in the system were sourced from the records of the Indian Government.

The selected parameter used for the study is listed below.

1) Rainfall (mm)
2) Maximum temperature (degree Celsius)
3) Crop Production (Tonnes)

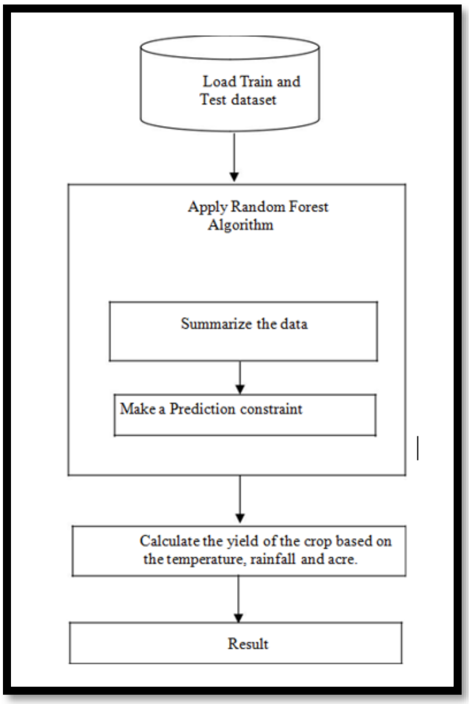


Fig. 2.8(a) Architecture of the proposed model.[13]

Algorithm Used: Random forest is a most popular and powerful supervised machine learning algorithm capable of performing both classifications, regression tasks, that operate by forming a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the particular trees. The more trees in a forest, the more robust the prediction. Random decision forests correct for decision trees habit of overfitting to their training set—the more trees in a forest, the more robust the prediction. Random decision forests correct for decision trees habit of overfitting to their training set.

According to this algorithm, convert the collected data sets into CSV file format and load those data sets. Split the loaded data sets into two sets, such as training data and test data in the split ratio of either 67 percentages or 33 percentages that is 0.67 or 0.33. To Separate the training data by class values so that the attribute map to a suitable value and stored in the appropriate list. Then calculate Mean and Standard Deviation for needed tuple and then summarize the data sets. Compare the summarized data list, and the original data sets calculate the probability. Based on the result, the largest probability produced is taken for prediction. The accuracy can be predicted by comparing the resultant class value with the test data set. The accuracy can range from 0% to 100%.

****

CHAPTER-3

HARDWARE AND SOFTWARE ELEMENTS

3.1 Overview:

An open-source IoT platform that runs on the ESP8266 and has hardware-based on the ESP-12 module. This chapter provides a detailed insight into the hardware as well as software components used in the final system. All the essential aspects of the datasheet of these hardware products are discussed in this chapter. The main hardware framework is built using node MCU and sensors that are connected to it. Measurement is an essential subsystem in any significant system, whether it may be a mechanical system or an electronic system.

A measurement system consists of sensors, actuators, transducers, and signal processing devices. The use of these elements and tools is not limited to measuring systems. The word sensors are widely used in association with measurement systems. A sensor is a device that provides usable output in response to a specified amount, which is measured. In simple terms, a sensor is a device that detects changes and events in a physical stimulus and provides a corresponding output signal that can be measured and recorded. The output signal can be any measurable signal and is generally an electrical quantity.

After the collection of data, the majority of the role is shifted towards software work. This chapter provides a detailed description of platforms and tools used to build mobile application and machine learning models for the data collected using the sensors of the slave units. The initial software part provides a detailed structure of the app by providing knowledge of the libraries and functions used to build it. The last half of the software part describes the working of linear and logistic regression.

3.2 Hardware components:

This part describes the pinouts and a brief description of the essential parameters from the datasheet of hardware components used in the project. The hardware components are broadly categorized as a master unit that consists of NodeMCU as a processor and a slave unit that consists of sensors.

3.2.1 ESP8266 NodeMCU:

ESP8266 is an integrated Wi-Fi chip in the industry that has the facility to integrate antenna switches, RF, power amplifiers, low noise amplifiers, filters, power management modules. It requires minimal external circuitry, and the entire solution, including the front-end module, is designed to occupy minimal PCB area. ESP8266 integrates an enhanced version of Tensilica's L106 Diamond series 32-bit processor with on-chip memory and Wi-Fi functionalities. Moreover, ESP8266 is often combined with external sensors and other application-specific devices through its GPIOs. It is a very user-friendly and low-cost device to provide internet connectivity to any project. The module has multiple capabilities of acting as an access point (one that can create hotspots) and as a station (can connect to another server using Wi-Fi). Hence it can fetch and consume data. It can fetch data from the internet using API's to access any information for the project. It also has an exciting feature that makes it easily compatible with Arduino IDE.

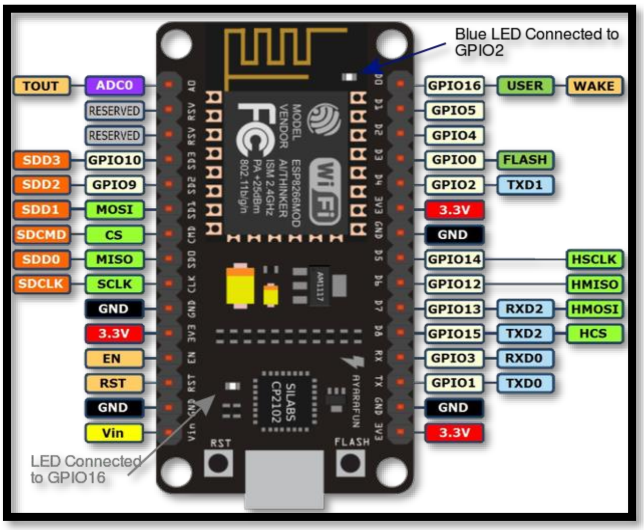


Fig 3.2.1(a) - NodeMCU Pinout

The pin wise description of the node is given by

1. **Power pin:** It has four power pins named as Vin and 3V3 pin. The Vin can be used to provide a direct supply to ESP8266 and its peripherals at a constant and regulated 5V. The 3V3 pins are the source to power the external components

2. **GND:** It provides the ground to the system.

3. **I2C pins:** These pins are to hook up all sorts of I2C sensors and peripherals in projects. Both I2C Master and I2C Slave are supported. I2C interface functionality can be realized programmatically, and the clock frequency is 100 kHz at a maximum. It should be noted that that the I2C clock frequency should be higher than the slowest clock frequency of the slave device. that I2C clock frequency should be higher than the slowest clock frequency of the slave device

4. **GPIO pins:** ESP8266 NodeMCU has 17 GPIO pins, which can be assigned to various functions such as I2C, I2S, UART, PWM, IR Remote Control, LED Light, and Button programmatically. Each digital-enabled GPIO can be configured to internal pull-up or pull-down, or set to high impedance.

5. **ADC channel:** The NodeMCU is embedded with a 10-bit precision SAR ADC. The two functions can be implemented using ADC viz. Testing the power supply voltage of the VDD3P3 pin and testing the input voltage of the TOUT pin. However, they cannot be implemented at the same time.

6. **UART pins:** ESP8266 NodeMCU has 2 UART interfaces, i.e., UART0 and UART1, which provide asynchronous communication (RS232 and RS485) and can communicate at up to 4.5 Mbps. UART0 (TXD0, RXD0, RST0, & CTS0 pins) can be used for communication

7. **Control pins:** Control pins are used to control ESP8266. These pins include Chip Enable pin (EN), Reset pin (RST), and WAKE pin.

- EN pin – The ESP8266 chip is enabled when the EN pin is pulled HIGH. When pulled LOW, the chip works at minimum power.
- RST pin – RST pin is used to reset the ESP8266 chip.

3.2.2 LDR:

A light-dependent resistor (LDR) is also known as a photoresistor or a cadmium sulphide (CdS) cell. This device works on the principle of photoconductivity, which states that a resistor whose resistance value decreases when the light intensity decreases. It consists of a snake-like track of Cadmium Sulphide (CdS) film that has a metal contact on both the terminal ends.

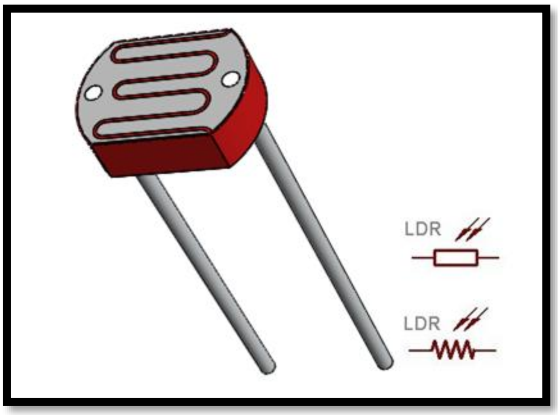


Fig 3.2.2(a) Light dependent resistor structure and symbol

Fabrication: Originally, obsolete materials like PbSe and PbTe were the earlier choice for the development of photoconductivity. However, Modern light-dependent resistors are manufactured from lead sulfide, lead selenide, indium antimonide, and most commonly cadmium sulfide and cadmium selenide.

LDR works on the principle of photoconductivity, which states that when the light falls on the surface of the sensor, the electrons in the valance band of the device are excited to the conduction band that reduces the metal conductivity. The conduction happens as the energy possessed by photons is higher than the bandgap of the semiconductor materials. Thus, when light falls on the LDR, then the resistance decreases and increases in the dark.

There are two types of LDR sensors Intrinsic photoresistors and Extrinsic photoresistors. The difference is based on the internal structure and composition of the material. In the case of the extrinsic photoresistor, the devices made from these materials are doped with impurities, which creates a new energy band above the valance band. The filling of electrons in the band decreases the gap and a small amount of energy is required in moving them. In the case of intrinsic photoresistors, they are pure semiconductor devices like silicon and germanium. When the light is incident on the LDR surface, then the electrons from the valance band to the conduction band and charge carrier increases tremendously.

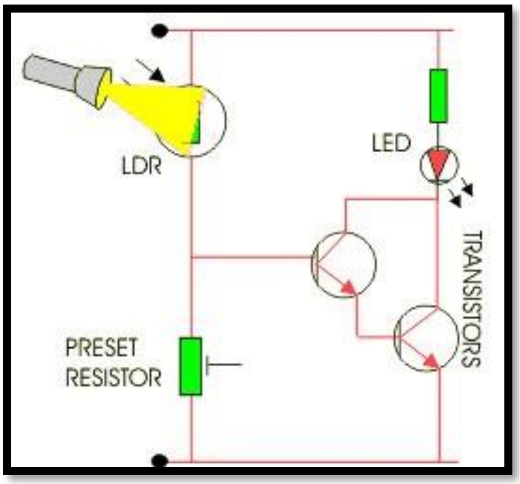


Fig 3.2.2(b) Internal circuit diagram of LDR

The circuit diagram of the LDR module consists of a pair of transistors connected as Darlington pairs. A voltage divider circuit connected across the base of the transistors, whose one end includes the light-dependent resistance and the other end consists of a fixed resistance to regulate the output voltage.

$$\boldsymbol{R = A.E^{a}}$$

The equation to show the relation between the resistance and illumination is given by

Where, E - illumination (lux)
R-resistance
A, a - constants

The value of 'a' depends on the CdS used in the manufacturing process. The amount generally lies between 0.7 and 0.9.

The graph shows the change in resistance with light intensity. It has a rectangular hyperbolic curve that shows a reduction in resistance with the increase in light energy. It will be useful in indicating theoretical resistance change with the light.

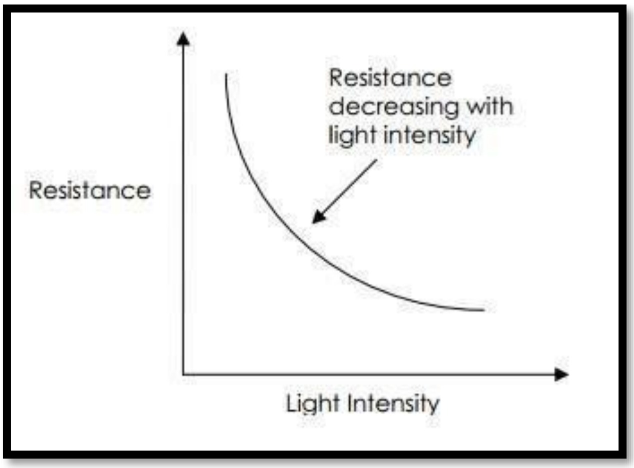


Fig 3.2.2(c) Resistance v/s intensity graph for LDR

3.2.3 Soil moisture sensor:

The sensor that measures water contents as an analog value and provides output in terms of percentage content is a soil moisture sensor. The water content between the soil particles is known as soil moisture. The moisture content is affected by many naturally occurring phenomena like precipitation and temperature change. However, the same factor significantly determines the presence of biome and land sustainability. The primary factor in deciding the growth and health of the plant is water or moisture content in the soil. The health crops rely upon an adequate supply of moisture and other nutrients.

The sensor is interfaced with NODE-MCU using the comparator module attached to it. The supply and ground of both the devices will be connected to each other. The analog pin of the node will be connected to the "A0" pin of the sensor.

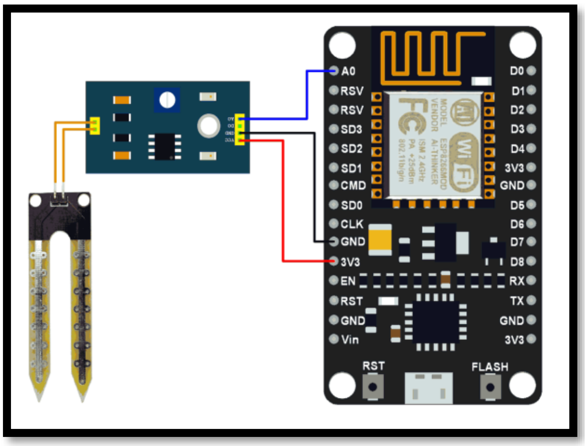


Fig 3.2.3(a) Node interfacing with soil moisture sensor

There are two types of sensors when it comes to moisture measurement:

- Soil Volumetric Water Content-based soil moisture sensors: These are applied to determine the amount of water present in the soil. It gives output in percent of the content.
- Soil Water Tension-based soil moisture sensors: These sensors measure the energy of water in the soil. The tension is measured in the strength of the earth. It tells how difficult or easy it will be for the plant to extract water from the soil.

A capacitive network that measures the dielectric permittivity of the surrounding medium to measure the moisture content. The dielectric permittivity of soil is the function of the water content. Hence, the sensor creates a voltage potential proportional to the dielectric permittivity that ultimately gives a direct proportion with water content. An average overall water content over the entire length of the sensor is being converted to an appropriate analog signal. This capacitive sensor provides a significant aspect of measurement, like loss of moisture content by the soil. Soil moisture sensor consists of two conducting plates that perform a combine function as a probe and a variable resistor. The insertion of the sensor in the soil causes the resistance to decrease for improving the conductivity between the plates.

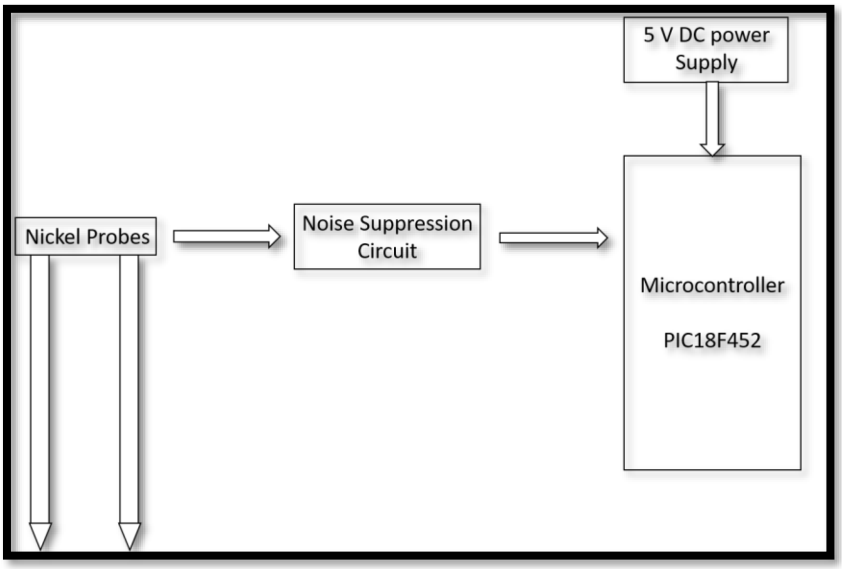


Fig 3.2.3(b) Block diagram for connection

The output will vary on the scale of 0-5 Volt based on the water content in the soil. In the case of zero moisture content, the circuit will act as an open circuit, and for the instance of full moisture content, it will conduct with zero approximately zero resistance. The sensor gives both analog and digital output. Moreover, it has a built-in potentiometer for sensitivity adjustment of the digital input; two sets of LEDs to indicate the connection and digital output of the sensor to detect the moisture content. The analog output varies between 0 and 1023 to report the moisture content, which is later converted to a scale of 0 to 100%. The processing of this sensor involves the use of ADC. The moisture content in terms of percentage is displayed on the serial monitor.

3.2.4 Temperature and humidity sensor:

Humidity is the measure of the amount of water vapor present in the air. The calculation involves terms like Relative humidity and Absolute humidity. For industrial and medical environments, relative humidity becomes an essential factor. A rise in the values of humidity, beyond threshold levels, can lead to malfunctioning of control systems, errors in weather prediction systems. So, as a security and safety factor, measurement of humidity values is crucial. Humidity sensors are used to measure the humidity values. Relative sensors also measure air temperature. However, this type of sensor is not useful for temperatures above 100 degrees Celsius.

Humidity Sensors are the low cost-sensitive electronic devices used to measure the humidity of the air. These are also known as Hygrometers. Humidity can be measured as Relative humidity, Absolute humidity, and Specific humidity. Based on the type of humidity measured by sensor, these are classified as Relative Humidity sensor and Absolute Humidity sensor. Based on the parameters used to measure humidity, these sensors are also classified as Capacitive Humidity Sensor, Resistive Humidity Sensor, and Thermal Conductivity Humidity Sensor.

A thermistor is a temperature-controlled resistor. The resistance offered by this reliable state temperature-controlled device depends on the ambient temperature. All resistors do have a temperature dependency that is given by their temperature coefficients. For most of the resistors (fixed and variable), this temperature coefficient is kept very low, such that the change in temperature does not significantly affect their resistance. On the other hand, the temperature coefficient of the thermistor is considerably high, thus their resistance change concerning a change in temperature.

A thermistor is mostly made from sensitive semiconductor-based metal oxides with metalized or sintered connecting leads onto a ceramic disc or bead. Thus, we can define a thermistor as: “A two-terminal solid-state thermally sensitive transducer that allows significant changes in its resistive value for the change in ambient temperature”.
DHT11 is an electronic instrument used to measure the temperature and humidity of the surrounding environment. It has a combined architecture of a capacitive humidity sensor and thermistor. It is a low-cost device that has an inbuilt analog to digital converter to provide excellent and precise results. It consists of four pins; each of them has different tasks to perform various tasks. For sensing the temperature, DHT11 uses an NTC temperature sensor, i.e., a variable resistor, which changes with the heat. For humidity sensing, a part of sensors consists of two electrodes having moisture-holding the substrate in between them. As the humidity level changes, the resistance between the electrodes also changes. Thus, the microcontroller reads this value and provides precise value. This system can be interfaced with node MCU with the digital pin to get the output of temperature and humidity.

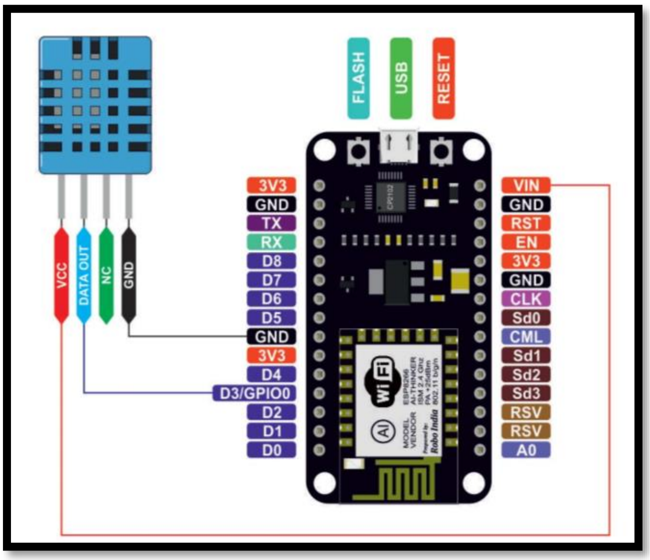


Fig 3.2.4(a) NodeMCU interfacing with DHT11

Relative humidity sensors usually contain a humidity sensing element along with a thermistor to measure temperature. For a capacitive sensor, the sensing element is a capacitor. Here the change in electrical permittivity of the dielectric material is measured to calculate the relative humidity values.

A selection of low resistive material is preferred for the construction of the sensor. It is kept on the top of the two electrodes, which reflects a change in the resistance with a change in humidity. Its temperature coefficient defined the temperature dependence of a resistor. According to this, the thermistors are classified into two categories based on the type of temperature constant. There are two types of coefficients, namely negative temperature and positive temperature coefficient. The ceramic semiconductor material used for each type of the thermistor, differs, as the temperature coefficient is dependent on the material used.

NTC or negative temperature coefficient thermistor is a device whose resistance decreases with an increase in temperature. These types of resistors usually exhibit a substantial, precise, and predictable decrease in resistance with an increase in temperature.

Characteristic Curve – A typical NTC thermistor gives the most precise readings in the temperature range of -55 C to 200 C. However, some specially designed NTC thermistors are used at absolute zero temperature (-273.15oC), and some can be used above 150-degree centigrade.

The figure below shows the characteristic curve of an NTC thermistor

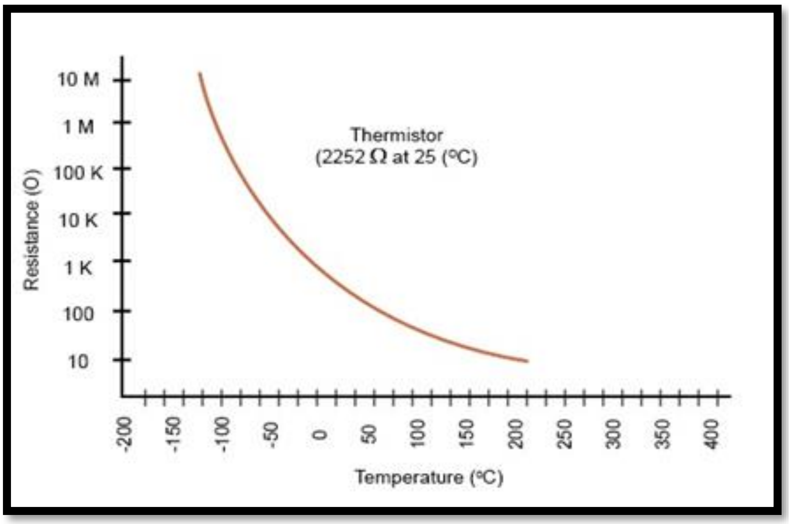


Fig 3.2.4(b) Resistance v/s temperature graph

It can be seen that they have a steep resistance temperature curve, denoting good temperature sensitivity. Out of all the approximations, the simplest one is:

$$\Delta R = k\Delta T$$

Where k is the negative temperature coefficient of the thermistor

The following equation can approximate the resistance and temperature relationship:

$$R = R' e^{\beta}\left(\frac{1}{T} - \frac{1}{T'}\right)$$

Where,

R = Resistance of Thermistor at the temperature T (in K)

R' = Resistance at given temperature T0 (in K)

β = Material specific-constant

In terms of temperature coefficient, the relation is given by:

$$R = R'[1 + \alpha(T - T')]$$

3.3 Software components:

This part describes the software tools used in the project. Moreover, it also provides a brief description of the libraries and algorithms used to get the results.

3.3.1 Firebase:

Firebase is a mobile and web application development platform that was initially created by Firebase Inc. in 2011. In 2014, Google acquired it. As of March 2020, Firebase offers 19 products that are being used by more than 1.5 million apps. Its vast range of products includes Firebase Authentication, Firebase Real-time Database, Firebase Hosting, Firebase Cloud Functions, and many more. This project uses various services provided by Firebase. These services are discussed in detail later.

Firebase is a Backend – as – a – service or BaaS. BaaS is a type of cloud service model where developers can outsource all behind the scene technicalities to these vendors. Thus, enabling developers to focus on creating and maintaining the frontend and client-side of their application without worrying about the server-side code. Firebase categorizes its services in 3 parts viz. Develop, Grow, and Earn. These services contribute to various stages of the system.

Firebase was integrated into our mobile application by adding the corresponding dependencies in the system. These dependencies are a type library that contains all the corresponding code related to Firebase, which can be used by merely calling those classes in our code. Different services of Firebase have different dependencies, which can be easily added in the system and used.

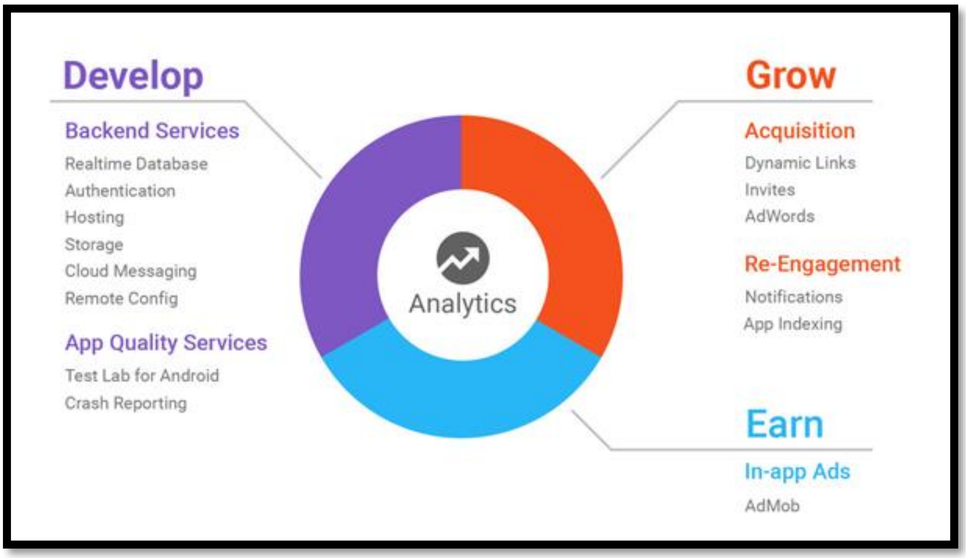


Fig. 3.3.1(a) Firebase Services

This project uses the following firebase services as listed below.

- Firebase Real-Time database
- Firebase Authentication
- Firebase Cloud Functions
- Firebase Cloud Messaging

Firebase has a real-time database that stores and provides all the data in real-time. This service provides developers with an API to synchronize application data across all clients and store on Firebase's cloud. This service can easily integrate with various types of applications such as Android, iOS, and Node.js.

Firebase also has libraries for integrating with hardware devices like Raspberry Pi and NodeMCU. Data collected by these hardware devices can quickly be sent to Firebase cloud using these libraries over the internet. These devices can also receive data from Firebase similarly. This facility makes Firebase a good option for cloud storage for IoT applications.

Firebase Real-Time Database is a NoSQL database. It allows storing and syncing data in real-time. Data is stored in a tree format in this database and is readily available in JSON format. Most of the cloud services provide data when requested for it, but Firebase provides data as and when it is updated. Thus, users get real-time data. A real-time database also includes security rules. These rules protect the data stored on Firebase, which ensures the safety of the user's data.

Firebase Authentication service authorizes the users using only client-side code. The service supports authorization via various platforms such as Facebook, Twitter, Google, GitHub, and many more. Apart from these options, the users also authenticate via email and password or even by their registered mobile numbers. Our project uses the email and password authentication system.

Firebase Authentication integrates directly into the Firebase Database. Thus, authenticated users can directly access the data as per the security rules set up by the system administrator. This service also includes a Firebase UI system that provides a customizable and open-source solution to handle sign-in UI flow for the end-user. Creating an authentication system is a very tedious and time-consuming job, but Firebase Authentication is an easy solution to this problem.

Every user who creates an account using this service gets a unique user id. This id identifies the user uniquely. Firebase Authentication also provides a way for the developer to access information of the user, such as user id or email id.

Cloud Functions provide the developer with a way to run the server-side backend code. This code is written in the form of functions. These functions trigger in response to various events. These events include Firebase features and HTTPS requests. These functions are written in JavaScript or Typescript languages. Cloud Functions save the developers from the hassle of scaling and managing their servers.

These functions can perform various tasks. These tasks are implemented on the server-side. Usually, computationally massive programs and tasks are written in the form functions. Doing so reduces the complexity as well as prevents heavy computations on the client-side. Many systems in today's technological world are across various platforms. Thus, having implementation or computation-based code on the server-side may save developers from implementing the same code multiple times.

Developers can use cloud functions to keep their users updated with relevant information by providing notifications. Cloud Functions are also used to clean and sanitize the database stored in the database. They are thus ensuring that the user always accesses relevant and useful information and also optimizes the database. The system may also use some third-party services. These types of services can be easily integrated with Firebase using Cloud Functions.

Firebase Cloud Messaging (FCM) is a cross-platform solution for sending messages and notifications across platforms such as Android, iOS, and web applications. It is an upgraded version of Google Cloud Messaging (GCM). As of now, it is available free of cost. Notifications can be sent to individual devices, groups of devices, or users subscribed to a particular topic of interest. FCM also supports sending messages from the client-side to the server. Acknowledgments, chat, or other messages can be sent from client devices to the server over FCM. Firebase Cloud Messaging also supports image as a payload in the notifications. The maximum payload size allowed for these notifications is 4 KB.

3.3.2 Android Studio

Android Studio is the official IDE (Integrated Development Environment) for Google's Android Operating System. It is based on JetBrains' IntelliJ IDEA software and is designed especially for Android development. Android Studio supports Java and C++ languages for application development. However, Kotlin has recently replaced Java as the preferred language for Application development. Android Studio is available for Windows, macOS, and Linux operating systems.

Android Studio performs a lot of tasks to automate the process of Application development. There are various types and categories of files and resources used to generate the application. Android Studio uses Gradle as a build management tool for integrating all various resources and deploying the application for use. Android Studio also supports building applications for Wearables, TV, and tablets. The current stable release of Android Studio (as of now) is 3.6. We have used version 3.5.3 of Android Studio for developing our application.

3.3.3 Other External Libraries Used:

Apart from all the other services and platforms, this system also uses two external libraries for developing the android application. These libraries have made adding features such as sharing the data in PDF format and representing variation in parameters such as humidity and temperature possible.

PDF files are a popular way of sharing data. These types of files are compatible across various platforms and can be printed out easily. Hence, it is quite useful to have such a facility to share data in a PDF file format. Apache iText is a popular library used to generate and manipulate PDF documents programmatically. This library was initially created for Java. Later it was ported to .Net, Android, and other platforms. The android version of iText is called iTextG. IText has open source as well as closed source components. This system uses the open-source components of the library.

This library provides classes to generate interactive PDF documents. Bookmarks, page numbers, and watermarks can also be added. PDF files can be split into two or more parts, and at the same time, various PDF files can be merged into one as well using this library. Using iText, we can save the PDF file as an image. In our project, we have used the iText library to generate a PDF containing the details of the farm. This file gives an overview of the current state of the farm.

Many times, we have to deal with a large dataset. In such a situation, it is useful to have a visual representation of data. MPAndroidChart is a popular Android library used for graphical representation of data. Line, bar, and pie chart can be quickly drawn using this library. This library is open source and freely available on GitHub.

Customizations in graphs are also supported. Colors and description text can be added to make the graphs more readable. Library also supports animations. These animations give a building up effect on the graphs

3.3.4 Linear regression algorithm:

A machine learning algorithm that models the relationship between one or more variables and allows mapping between numeric inputs and outputs by fitting a regression line into the data points. Linear regression is an essential statistical tool that determines the relationship between two variables. By using linear regression, we can check the strength of the correlation between two or more variables and the contribution of each variable in determining the appropriate results. Linear regression is generally used to predict the target variable y using the regression plot that is drawing using the training data.

The working method for this algorithm can be describes as: for a given data set {Yi1, Xi1, Xi2, Xi3, ..., Xip} for i=0 to n (rows), a relationship between the dependent variable and the p-vector of input constitutes the regression model. The association is modelled utilizing error methods. This error acts as an unobserved random variable that adds noise to the linear relationship between the dependent and independent variables.

The model generally takes the form of the equation denoted by:

1. A single variable linear regression model

$$y_i = \alpha + \beta x_i + \varepsilon$$

2. A multi-variable linear regression model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

where,
y: It is the vector of observed values Yi known as regressand or dependent variable.
x: It is the vector of input variables Xi known as regressors.
α or βo: is the intercept form of regression coefficients.
βi: It is a vector of regression constant

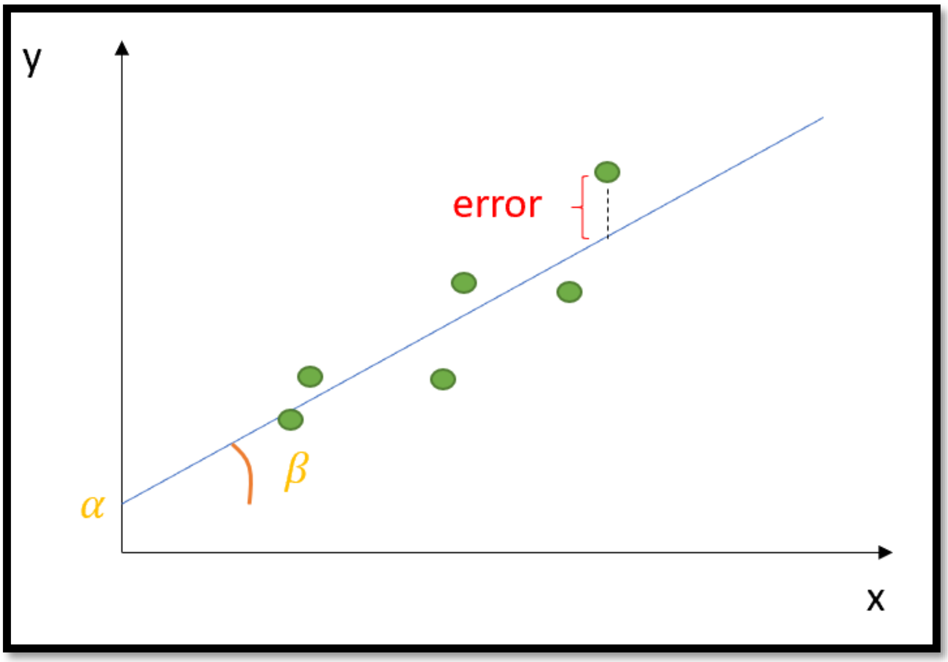


Fig 3.3.4(a) Error between regression line and datapoint [20]

To find the coefficients based on the dataset, we minimize the least-squares to the sum of squared errors. The linear model built using this assumption will not be perfect, and it will not predict the data accurately as the error function will give the difference.

The idea of Simple Linear Regression is to find regression coefficients for which we minimize the error term. Precisely the model works to keep the squared error minimum. The reason for using the least square is that both positive and negative values of the dataset should be equally penalized for error. The figure shows a data model that fits linear regression as a machine learning algorithm. The graph is plotted between variables X and Y. [20]

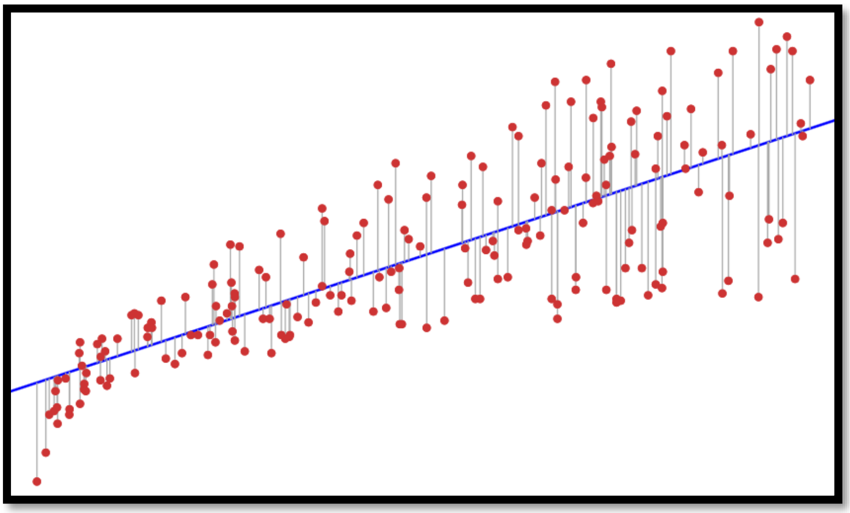


Fig 3.3.4(b) Error between regression line and all the points in the dataset

The error is calculated between the different possible regression lines and data points. Then the square of the error is minimized to check the best-fitting regression line through the data. The blue color line shows the optimized regression line that will form the equation to predict the test data's target variables. The coefficients for single variable regression can be calculated using the following correlation method that finds the value of βi. Using βi and the average amount of input and output variable, we can calculate βo or α

$$\sum_{i=1}^{n}\left(y_i - \hat{\alpha} - \hat{\beta} * x_i\right)x_i = 0$$

$$\Rightarrow \sum_{i=1}^{n} y_i x_i - \hat{\alpha} x_i - \hat{\beta} x_i^2 = 0$$

$$\Rightarrow \sum_{i=1}^{n} y_i x_i - \left(\bar{y} - \hat{\beta}\bar{x}\right)x_i - \hat{\beta} x_i^2 = 0$$

$$\Rightarrow \sum_{i=1}^{n} y_i x_i - \bar{y} x_i + \hat{\beta}\bar{x} x_i - \hat{\beta} x_i^2 = 0 \quad \Rightarrow \quad \sum_{i=1}^{n}\left(y_i - \bar{y} + \hat{\beta}\bar{x} - \hat{\beta} x_i\right)x_i = 0$$

$$\Rightarrow \sum_{i=1}^{n} y_i - \bar{y} + \hat{\beta}\left(\bar{x} - x_i\right) = 0 \quad \Rightarrow \quad \sum_{i=1}^{n}\left(y_i - \bar{y}\right) = -\hat{\beta}\sum_{i=1}^{n}\left(\bar{x} - x_i\right)$$

.

$$\Rightarrow \hat{\beta} = \frac{\sum_{i=1}^{n}\left(y_i - \bar{y}\right)}{\sum_{i=1}^{n}\left(x_i - \bar{x}\right)}$$

$$\Rightarrow \hat{\beta} = \frac{\sum_{i=1}^{n}\left(y_i - \bar{y}\right)\left(x_i - \bar{x}\right)}{\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^2}$$

The model calculates the coefficients using the curve fitting techniques but is the model relevant to the data and how well has the model fitted to the dataset can be estimated by p-value and the null hypothesis. The p-value quantifies statistical significance by allowing us to tell whether the null hypothesis is to be rejected or not. The low variance of the data is a useful characteristic to fit the regression line properly. For any model, the hypothesis assumes that there is some correlation between the target variables and feature variables. The null hypothesis acts opposite, stating that there is no correlation between the variables. Therefore, by finding the p-value for each coefficient we can determine whether the variable is statistically significant to predict the target. In general, the p-value, which is less than 0.05 indicates a strong relationship between the variables. There are various parameters which are calculated in order to find the accuracy of the model, which are given by:

(RSS) Residual sum of Squares - $\sum[\text{Actual}(Y) - \text{Predicted}(Y)]^2$

(ESS) Explained Sum of Squares - $\sum[\text{Predicted}(Y) - \text{Mean}(Y\text{mean})]^2$

(TSS) Total Sum of Squares - $\sum[\text{Actual}(Y) - \text{Mean}(Y\text{mean})]^2$

The various parameters that can assess the model can have different values based on the type of dataset. The regression analysis has the following metrics to evaluate the model:

1. **R Square (Coefficient of Determination)** -This metric explains the percentage of variance that ranges between 0 and 1. A higher value is expected, but it rests on the data quality and domain.

$$RSE = \sqrt{\frac{1}{n-2}RSS} = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS}, \quad TSS = \sum (y_i - \hat{y}_i)^2$$

2. **F Statistics** - It estimates the overall significance by calculating the ratio shown in the figure that compares the fully trained model with an intercept only (no predictors) model. The value of this parameter can range between zero and any arbitrarily large number.
3. **RMSE / MSE / MAE** - Error based parameters are the crucial evaluation number lower the amount, better the model.
   - **MSE** - This is mean squared error. It tends to magnify the impact of outliers on the model's accuracy.
   - **MAE** - This is a mean absolute error. It is firm against the effect of outliers.

RMSE - This is the root mean square error. It is interpreted as how far, on average, the residuals are from zero. It invalidates the squared effect of MSE by square root and provides the original units as data. [19]

3.3.5 Logistic regression algorithm:

A classification algorithm that assigns observations to a discrete set of classes is known as logistic regression. It utilizes the logistic sigmoid function to create a decision boundary between two distinct classes and returns a probability value.

There are two types of logistic regression algorithms:

1. Binary
2. Multi-linear functions

It is a predictive analysis algorithm based on the concepts of probability. Logistic regression uses a complex cost function based on the sigmoid and linear relationship of the variables to create a decision boundary that determines the likelihood of class to which it can belong. The hypothesis of logistic regression is to limit the cost function between 0 and 1. Therefore, we need to change the Mathematical function from linear to sigmoid.

We need to modify the function in the following manner and plot the resulting sigmoid curve.

The linear function is given as,

$$h\Theta(x) = \beta_0 + \beta_1 X$$

For logistic regression, it is transformed as

$$\sigma(Z) = \sigma(\beta_0 + \beta_1 X)$$

The final cost function for the binary logistic classifier is given by

$$Z = \beta_0 + \beta_1 X$$

$$h\Theta(x) = \text{sigmoid}(Z)$$

$$\text{i.e. } h\Theta(x) = 1/(1 + e\text{^-}(\beta_0 + \beta_1 X)$$

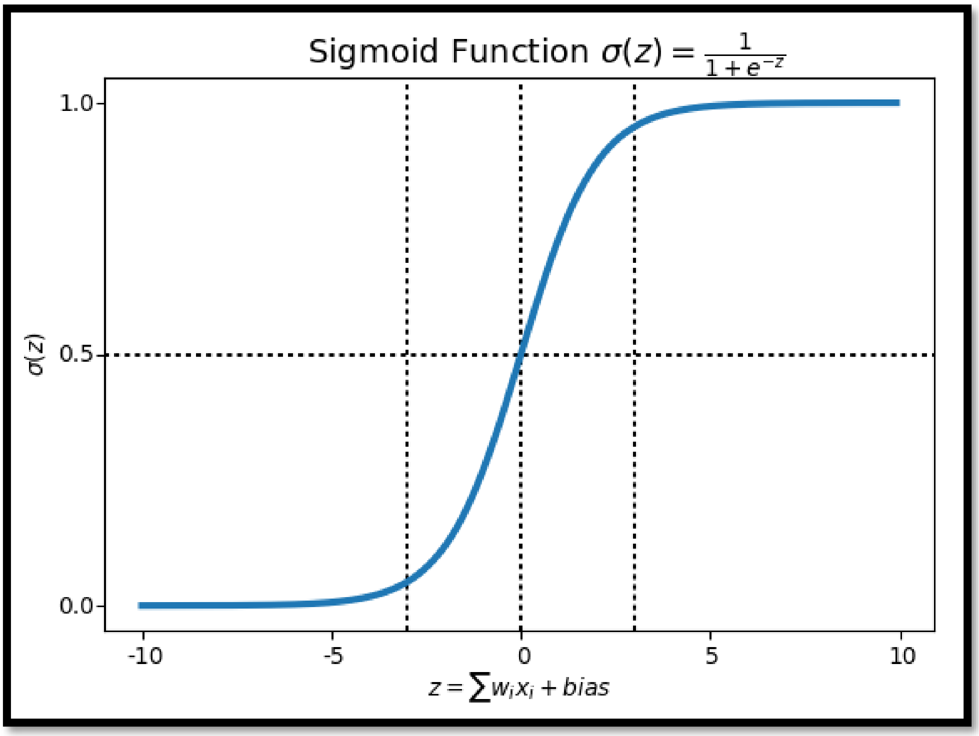


Fig 3.3.5(a) Sigmoid function [20]

The classifier should act in such a way that the set of outputs received from the prediction function should return a probability score between 0 and 1. In order to determine the maximum-likelihood estimation of your classification, the coefficients of logistic regression must be estimated from the training data set.

$$h\theta(X) = \frac{1}{1 + e^{-(\beta_o + \beta_1 X)}}$$

Maximum Likelihood estimation a shared learning technique used by a variety of machine learning algorithms. However, it does make assumptions about the distribution of your data. The best coefficients would result in a model that would predict a value that is very close to 1 for the default class, and value is very close to 0 for the other type. The intuition for maximum-likelihood for logistic regression is that a search procedure seeks values for the coefficients, which act towards minimizing the error in the probabilities predicted by the model to those in the data.

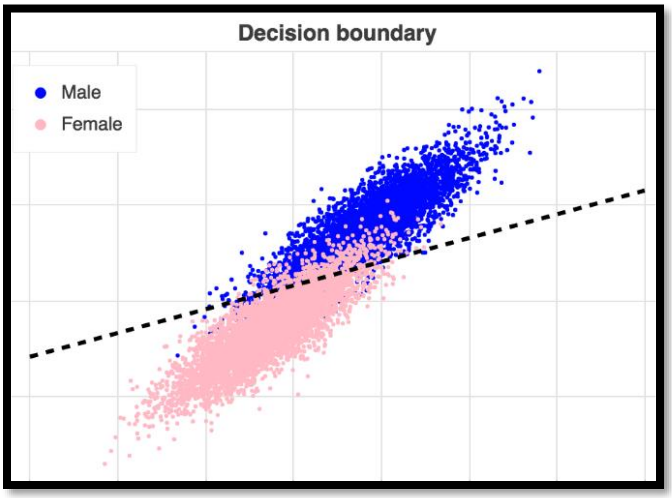


Fig 3.3.5(b) Decision boundary for classification [20]

Making predictions with logistic regression is a simple task of putting the numbers in the sigmoid equation with the regression coefficients. For instance, if we place B0, B1, and the value of input for which we need an answer. We can substitute the values to get the answer for Y.

ADVANTAGES

- One of the simplest machine learning algorithms yet provides excellent efficiency.
- Variance is low.
- It can also be used for feature extraction.

- Logistic models can be updated easily with new data using stochastic gradient descent.

- Logistic regression operates all different sets of metrics compared to linear regression. In this case, we deal with probability and categorical variables. These are the metrics for assessing a logistic regression model

1. **Akaike Information Criteria (AIC):** It is a significant indicator of model fit, which follows the rule: Smaller the better. AIC penalizes an increasing number of coefficients in the model. The model with relatively low AIC will be better.

2. **Null Deviance and Residual Deviance:** Deviance calculation is carried out as -2 times of log of the likelihood of an observation. There are two types of

deviance: Null and residual deviance. The first one can be calculated using a model with no features to predict intercept (by constant probability). The second one is calculated with all the features. The wider the difference between null and residual deviance, the better the model.

3. **Confusion Matrix:** The most commonly used metric to evaluate classification models. The structure of the confusion matrix is shown in the diagram below.

	1 (Predicted)	0 (Predicted)
1 (Actual)	True Positive	False Negative
0 (Actual)	False Positive	True Negative

Fig 3.3.5(c) Confusion matrix structure [19]

It utilizes a tabular format to avoid confusion by measuring true and predicted values. A confusion matrix can interpret the following parameters.

- **Accuracy** - It determines the overall accuracy of the predicted variables model. It is calculated as Accuracy = (TP (True Positives) + TN (True Negatives))/ (TP + TN + FP (False Positives) + FN (False Negatives)).

- **True Positive Rate (TPR)** - It indicates the correctly predicted positive values, out of all the positive values. The formula to calculate the true positive rate is (TP/TP + FN). Also, TPR = 1 - False Negative Rate. It is also known as **Sensitivity** or **Recall**.
- **False Positive Rate (FPR)** - It indicates the incorrectly predicted negative values, out of all the negative values. The formula to calculate

the false positive rate is (FP/FP + TN). Also, FPR = 1 - True Negative Rate.

- **True Negative Rate (TNR) -** It indicates the correctly predicted negative values, out of all the negative values. The formula to calculate the true negative rate is (TN/TN + FP)
- **False Negative Rate (FNR)** - It indicates the incorrectly predicted positive values, out of all the positive values. The formula to calculate the false-negative rate is (FN/FN + TP).

4. Receiver Operator Characteristic (ROC): It is a user-defined threshold method to determine the accuracy of classification by the use of AUC (Area under the curve) method.

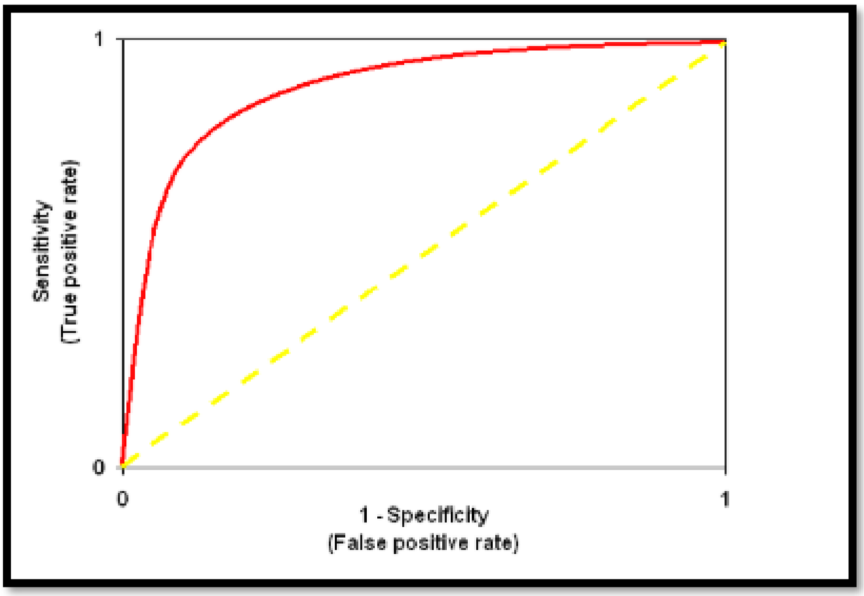


Fig 3.3.5(d) Receiver operator characteristics [19]

The area under the curve (AUC), also referred to as an index of accuracy (A) or concordant index, represents the ROC curve's performance. Higher the area, better the model. ROC is plotted between the True Positive Rate (Y-axis) and False Positive Rate (X-Axis).

****

CHAPTER-4

SYSTEM DESIGN AND ARCHITECTURE

4.1 Overview:

Previous sections gave an introduction to the system's domain as well as the system's components. This section focuses on the design and architecture of the system we have made. The proposed system has been divided into four sections. The first section deals with the hardware used in the system. This hardware contains the sensors which collect the data and stores it to the firebase database, the cloud storage we have incorporated in our system. It saves the data collected by the hardware unit. The third section is about the user interface through which the user can access this data. The user interface includes an Android-based mobile application and a web application. Machine Learning used to analyze data and predict many useful results forms the fourth section.

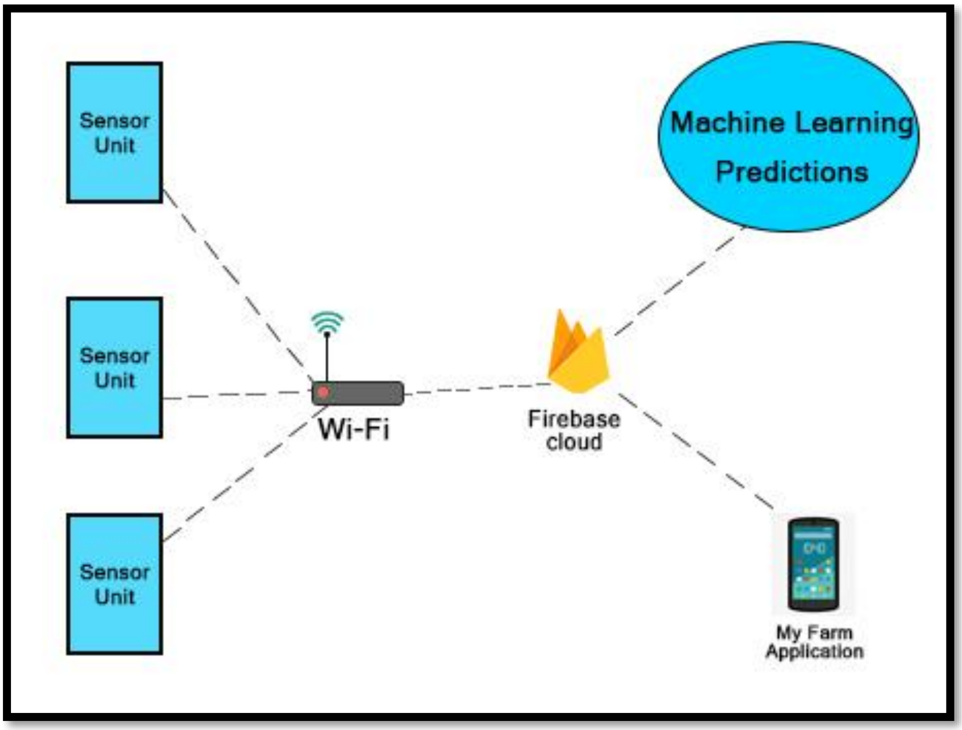


Fig 4.1(a) Overall system architecture

4.2 Hardware Unit

The proposed system consists of a hardware setup that collects data from the user's farm. This data is the readings obtained by various sensors. These readings reflect on the physical parameters of the farm, such as temperature, humidity, soil moisture, and light intensity. The system uses DHT11 as temperature and humidity sensor, LDR sensor to measure light intensity, and YL-69 soil moisture sensor to measure the soil moisture level.

These sensors measure the corresponding physical parameters of the environment and soil conditions. These sensors are connected to a NodeMCU using an inbuilt Wi-Fi module. Hence, it quickly gets connected to the Internet via Wi-Fi. Many libraries are freely available, which enables us to perform various tasks. NodeMCU has a readily available library for Firebase as well. This library makes it extremely easy to transfer and receive data to Firebase cloud storage.

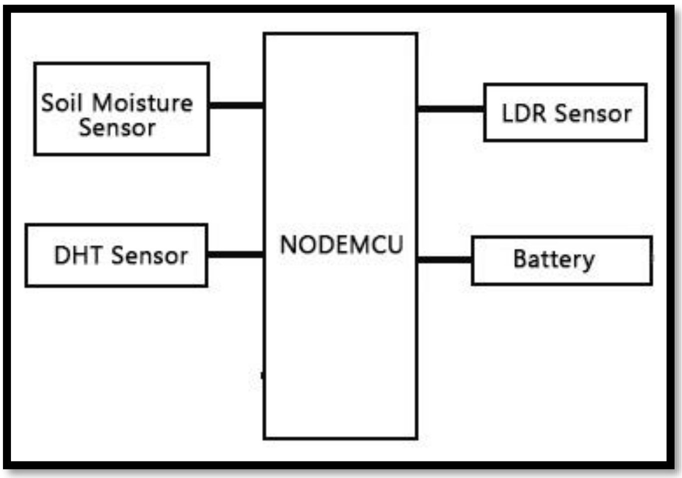


Fig 4.2 (a) Block diagram of Hardware unit

All the sensors and NodeMCU together form one hardware unit. The farm may contain more than one such hardware units. These units are placed at various locations throughout the farm, making it possible to get readings from different parts of the farm. Hence, a more general estimate of the farm can be made. Also, many times farmers prefer to go for "Mixed Farming". Different crops are planted in different parts of the farm. In such a situation having readings from various locations on the farm would be helpful. All units communicate data wirelessly, which ensures that there is no need for any wiring throughout the farm. Hardware units can be placed at the required location, and it is good to go.

The geographical area that these units can cover collectively depends on coverage on the range of Wi-Fi modem. A relay type of arrangement can be used to place units beyond the coverage range of the Wi-Fi modem. A buffer unit which can communicate wirelessly via the NRF24L01 module receives the readings from groups beyond the coverage range of Wi-Fi and transmits them to the Firebase database via Wi-Fi. Hardware units placed beyond this coverage range would also need an NRF24L01 module to communicate. A similar relay network can be used to increase the coverage area of the system further.

Every hardware unit deployed on the farm has a unique ID associated with it. This ID makes it possible to identify the particular group in the network and the Cloud database. As the system has multiple users, this ID given to each unit ensures that data from the unit reaches the correct address in the database. This ID is also helpful in tracking down the group in case some fault arises.

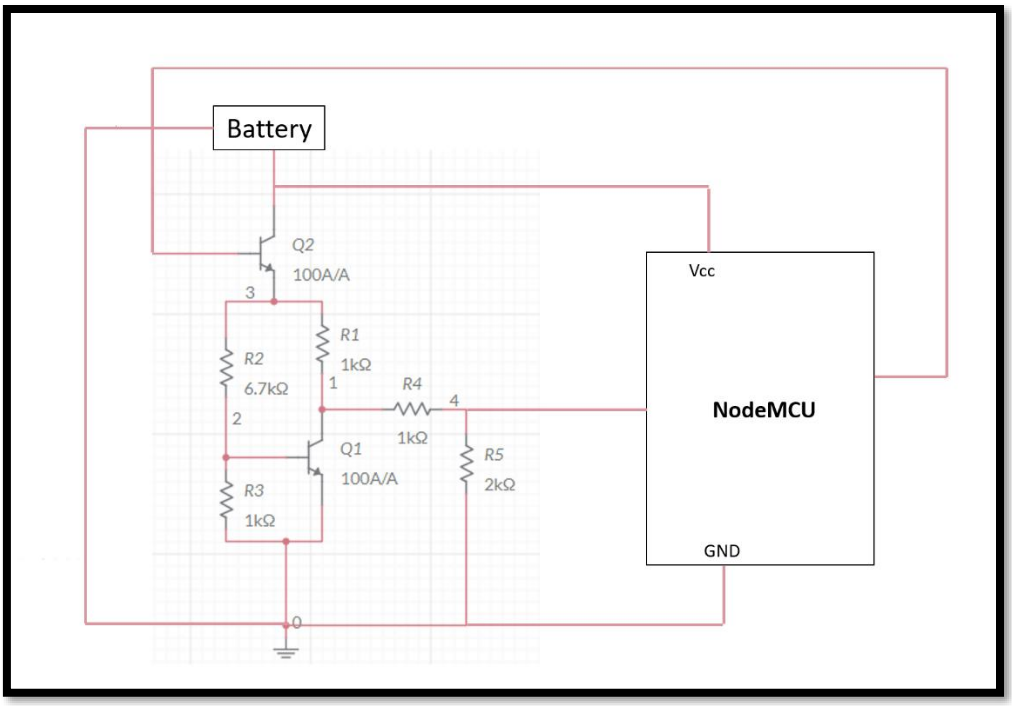


Fig 4.2 (b) Circuit diagram for low battery level indicator

The hardware units deployed on the farm are battery operated. The standard operating voltage level for NodeMCU is 7-12 volts. Thus, it becomes necessary to replace the battery when it gets discharged below a certain voltage level (6 volts generally). Keeping track of battery levels of all the units becomes overwhelming. So, the groups have an additional circuit to keep a check on the battery level. Transistor Q2 is added in the circuit to avoid continuous flow of current through the indicator circuit.

In the circuit shown in the above figure, the transistor Q1 turns off when the supply voltage goes below 6v. The resistors R4 and R5 form a voltage divider circuit. This circuit ensures that the voltage level at the input pin of NodeMCU does not exceed 5 volts. As soon as the voltage level of the battery drops below the set threshold, it alerts the farmer by sending a notification. The unique ID given to the unit is used here to tell the farmer exactly which group needs a battery replacement. Alternate sources of energy like Solar Power could also be used but using those energy sources for the units is beyond the scope of this project.

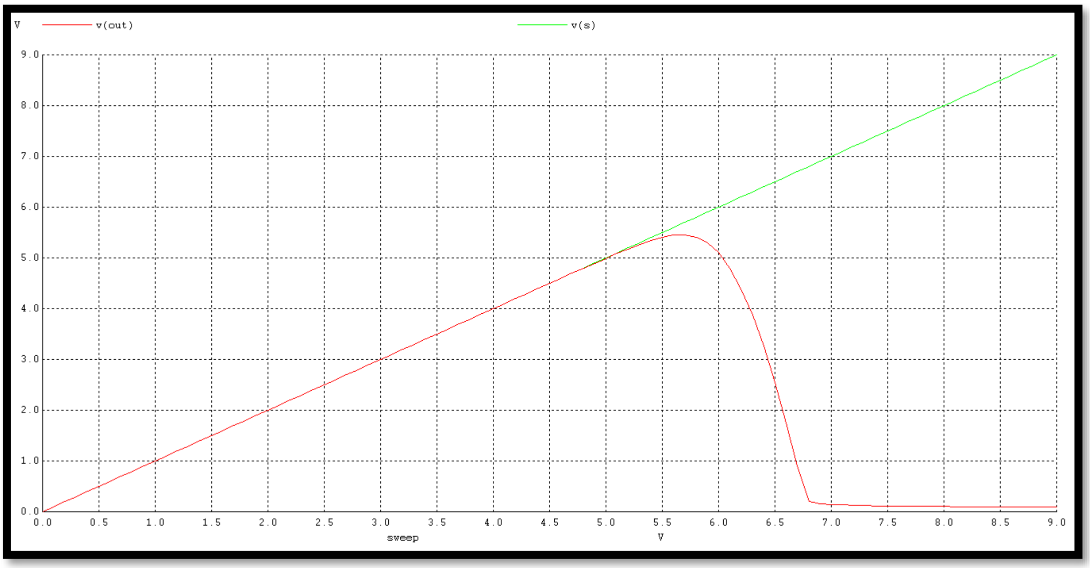


Fig 4.2 (c) Simulation graph for low Battery level indicator circuit

The parameters that are being measured by the system do not vary rapidly. Hence, the system takes the readings once an hour. Parameters such as temperature may not change every hour as it would be useless to have the same texts stored in the database repeatedly. This would lead to inefficient use of cloud storage space as well as communication bandwidth. Thus, the system is designed to ignore similar consecutive readings. Readings may be taken every hour, but they are transmitted to cloud storage only if a change in the reading is detected. Thus, some filtering of raw data is done at the basic hardware level itself.

4.3 Firebase Database

The data collected from various sensors is sent to Firebase's real-time database over Wi-Fi, which works as the cloud storage for this system. Readings obtained from different hardware units are stored in Firebase's real-time database. As a result, the farmer is always updated with the latest data even without refreshing. Firebase has many supporting libraries. The hardware support library makes it possible for readings from sensors to be stored in the database. Similar libraries allow data from Firebase to be displayed across various platforms such as Android or web.

4.3.1 Structure of the database

Similar to the unique ID given to each hardware unit, each user is also assigned a unique ID that identifies the user in the database. Data on Firebase is stored in tree format with the user ID of the user as the root node. Under this node, we have stored various information about the user. This information includes the name and contact number of the user. The number of devices currently deployed in the farm is also stored. Under the node “Devices”, the IDs of all the used units are stored. The node “Average” has four sub-nodes. Each of these sub-nodes stores the average value of all the obtained readings of the corresponding parameter. The node “Zones” stores the data from various individual units. Under Zones node, we have separate nodes for multiple units. These inner nodes contain the readings obtained from the corresponding group.

Along with readings, we also have a description associated with each hardware unit. This description describes the location where the unit is placed on the farm. In this way, the farmer can quickly locate the corresponding unit. The battery level of each unit is also monitored here. As soon as the voltage level goes below the fixed threshold, change is reflected in the database. Also we have stored a message token for user. This token is used to send notification to the correct device.

Everything stored in the database is easily sharable in JSON (JavaScript Object Notation) format. JSON is an accessible data format which used to transfer data over the Internet. JSON is easily readable in most of the present-day programming languages like Java and Python that makes sharing this data across various platforms extremely easy.

The system supports multiple users. Thus, users need to create their account to use the system. Firebase provides the service of authenticating the user. This system authorizes the user based on email ID and password. A new user can create his/her account using an email ID and password.

Fig 4.3.1(a) Structure in which data is stored on Firebase

As mentioned earlier, every hardware unit and every user is assigned a unique ID. When the hardware is configured, the user ID of the user is registered against the corresponding ID of the hardware unit, ensuring that the hardware unit knows precisely which user it has to send data. This device is also registered under the Devices node inside in the corresponding user's node. Thus, we can quickly get the list of units currently deployed in the particular user's farm.

Fig 4.3.1(b) User ID registered against Hardware ID

4.3.2 Processing of Data

The data received from the sensors is simply the numerical value of the reading of a physical parameter. These readings provide more useful information when associated with time. Thus, to tap the time when these readings are taken, we have configured Firebase to append a timestamp to all the readings that are stored. A Cloud Function named “AppendTime” is created for this purpose. This function is triggered when a new parameter reading is added at the corresponding node of the database. The service reads the value from the database, appends the current timestamp to it, and then stores the result at the same location again.

Readings from all the units are stored in the database. The obtained value is the average value of the corresponding parameter from all the units. The earned value is again appended by the corresponding timestamp to associate the value with time. The generated value is stored under the corresponding parameter node under the Average node in the database.

The average of these readings is calculated by a Cloud Function named “calculateAverage”. This function is triggered when a new parameter reading from any node is added to the database. The updated average value is calculated using the statistical method used for correcting averages. The total sum is calculated by multiplying the older average value with the number of devices. Earlier reading of that unit is subtracted from the aggregate, and newer reading is added. The result is again divided by the number of devices. Thus, in this way, we can efficiently update the average value without needing to fetch readings from all the individual nodes. The time complexity of this function becomes independent of the number of units deployed.

$$\text{New Average Value} = \frac{(OldAverage\ Value * NumberOfDevices\ ) - OldReading + NewReading}{NumberOfDevices}$$

The above equation fails to provide the expected result in one case. When we are going to have the first reading from a newly configured hardware unit, the output may not be as expected. This is due to the fact that the number of devices has changed. Thus, the sum calculated by multiplying older average value with the number of devices is incorrect. In such a situation, the average is calculated by the equation shown below.

$$\text{New Average Value} = \frac{(OldAverage\ Value * (NumberOfDevices - 1)\ ) - OldReading + NewReading}{NumberOfDevices}$$

The readings obtained from sensors may not be of desired value always. If any reading is found to be out of bounds, i.e. not according to the standard threshold values, the user is notified immediately. For e.g., if the soil moisture level goes below the standard threshold value, the farmer upon receiving this information can start the irrigation pump. Sending of such notifications is controlled by a Cloud Function named “notifyUser”. This function triggers in a similar way as “AppendTime” and “calculateAverage” did.

Upon receiving a reading from the sensor, it checks if the reading is above the threshold or not. If not so, then the farmer is notified using Firebase Cloud Messaging (FCM) system. FCM is Firebase’s built-in messaging and notification system. The function “notifyUser” triggers the FCM, and a notification is sent to the user. Every registered user gets a unique message token assigned by Firebase. This token tells Firebase about the device on which the notification should be sent. This notification contains details about which parameter in the corresponding unit has reported an inappropriate reading.

4.4 User Interface

The sections till now covered the collection and processing of the data. It is essential that the farmer can access this data for which a User Interface (UI) is required, which can display all the data for the farmer to view. This User Interface is provided in the form of an Android-based mobile application named “My Farm”.

The mobile application provides a decent UI for the farmer to view the data. This application is developed using software called Android Studio. XML is used to design the layout or the external UI of the App. Whereas Java is used to code the business logic of the App. Gradle is used here as a build management tool. It takes all the files of the App and bundles it into an APK file. This APK file is then installed in the mobile device to get the app running.

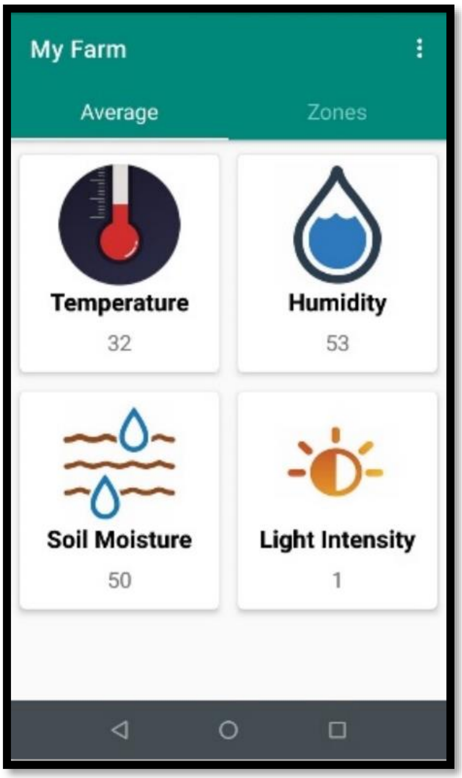
Android Studio needs to be configured to add Firebase in the system to receive a date from Firebase that is done by adding Firebase dependencies in a file called “build.gradle”. The corresponding dependencies added in this file provide the relevant Firebase services. After adding these dependencies, they behave similarly to external libraries. All the code is accessible by merely calling the corresponding functions in our system.

One of the most significant advantages of Firebase is real-time access to data stored in its database. There is no need to refresh or restart the application. This facility is possible by the use of a java class called Listener. Listeners are a type of class in Java that runs continuously in the background. Their job is to listen for a specific event for which they have been programmed. Once any such event occurs, they trigger the corresponding code according to the event. Firebase uses one such Listener to listen to the events of any changes in data. Once some change is detected in the stored data, the Listener triggers the corresponding code. This Listener is available in the Firebase Database library, which is added as a dependency.

The Listener we have used in this project is “ChildEventListener,” which is triggered when a stored value is changed or updated, or any new data is added. This Listener has been attached to a DatabaseRefrence object, which points to the location in the cloud database. This Listener listens for events with the change, update, removal, or addition of data. This Listener updates the data displayed on the App when it is triggered. Here, we have programmed the Listener to update the data in the application only when there is an addition of data in the database.

Users can access the data stored in Firebase after signing in the system. A new user can easily create his/her account and then configure the corresponding hardware unit to start using it. The login screen appears when the user installs and opens the app for the first time.

The application provides a decent UI for displaying the data. The screen is divided into two parts, where the first one shows the current average value of all the readings, and the second part shows readings from individual units. Clicking on any of these parameters, open a new screen (known as Activity), which displays all the previous average readings of the parameter. Selecting any of the units from the second part takes the user to a screen.



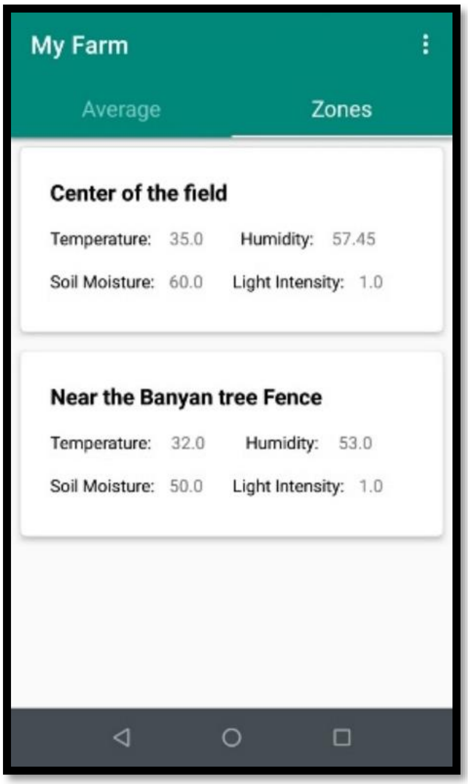


Fig 4.4(a) 1) average readings page 2) Individual Unit page 3) Past readings with variation graph

All these previous readings are displayed along with the date and time when they were reported. The Activity, which shows the previous readings, also represents the data in a graphical form. A line graph is plotted from the readings. This graph shows the variation of the corresponding parameter with time.

The average value of readings of various parameters shows the current state of the farm. Hence, it would be useful if this data could be shared. This App has the feature to share these details in a PDF file format. PDF (Portable Document Format) is an accessible and compatible file format for sharing data over various platforms. This feature also helps the farmer to ask for help from some experts regarding his farm. The PDF file includes the current average values as well as contemporary readings from various individual units. This file can be easily shared over various platforms such as WhatsApp, Google Drive or E-mail.

Fig 4.4(b) Application in local language Hindi

In a country like India, many farmers may not be comfortable using the App in English. Hence, the application is designed to support various local languages as well. The farmer can visit the setting page in the App and change the language to the one in which he is most comfortable. This feature ensures that language won't be a barrier in using our system.

The application includes translation files for various languages. The code is written in the settings page to set all the text in the application according to translation file of the language chosen by the user.

Thus, the following are the principal features provided by this mobile application:

- Latest data in real-time displayed in a decent UI.
- Displays aggregate data of the entire farm

- Easy configurations for adding a hardware unit.
- Graphical representation data to study the variation of a particular parameter over time.
- Facility to share the details of the farm in PDF file format.
- Regional Language Support
- Receives notification when some parameter goes beyond the threshold value

These features make this application extremely user friendly and easy to use.

****

CHAPTER-5

WORKING AND RESULTS

5.1 Data collection from hardware:

Time	Soil Moisture(percentage)	Current humidity	Temperature(C)
12:00:05	22	55	38
15:29:25	54.86	58	39
15:29:35	28.17	58	39
15:29:45	24.95	58	39
15:29:55	23.87	58	39
15:30:01	23.48	58	39
15:30:11	23.29	58	39
15:30:21	23.19	58	39
15:30:31	23.19	58	39
15:30:41	23.19	58	39
15:30:51	22.99	58	39
15:31:01	23.29	58	39
15:31:11	23.48	58	39
15:31:21	23.97	58	39
15:31:31	23.58	58	39
15:31:41	23.97	58	39
18:15:33	30.15	63	31
18:15:43	30.94	63	31
18:15:53	30.94	63	31
18:16:03	30.65	63	31
18:16:13	30.15	63	31
18:16:23	30.35	63	31
18:16:33	30.15	63	31
18:16:43	30.35	63	31
18:16:53	30.65	63	31
18:17:03	31.05	63	31
18:17:13	31.45	63	31
18:17:23	31.05	63	31
18:17:33	30.51	63	31
18:17:43	31.97	63	31
18:17:53	32.11	63	31
23:56:29	40.39	68	28

23:56:39	40.78	68	28
23:56:49	40.98	68	28
23:56:59	40.88	68	28
23:57:09	40.59	68	28
23:57:19	40.49	68	28
23:57:29	40.69	68	28
23:57	40.88	68	28
23:57:49	40.78	68	28
23:57:59	44.3	68	28
23:58:09	44.7	68	28
23:58:19	44.6	68	28
23:58:29	44.89	68	28
23:58:39	44.89	68	28
23:58:49	44.7	68	28

Table 5.1 Practical values of physical parameters

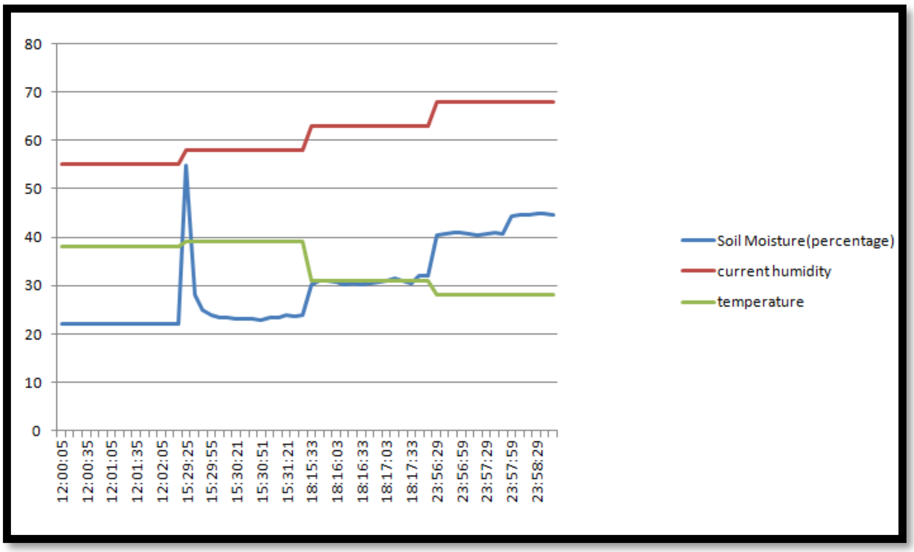


Fig 5.1(a) Graph of sensor readings over time

5.2 Work flow of mobile application:

This report has covered the design and architecture of the system until now. This section walks you through the working and the user flow of the system, starting from when the Android Application is installed. This section of the report can be considered as the documentation of the system.

The system uses an Android application named "MyFarm" as the user interface. The App displays all the data for a particular farm. When the user installs the Application for the first time, he/she is prompted to sign in. If the user is new to the system, he can create a new account. User can quickly sign in using his/her Email ID and password. In order to create a new account, the user has to provide some details, which include the name and contact number.

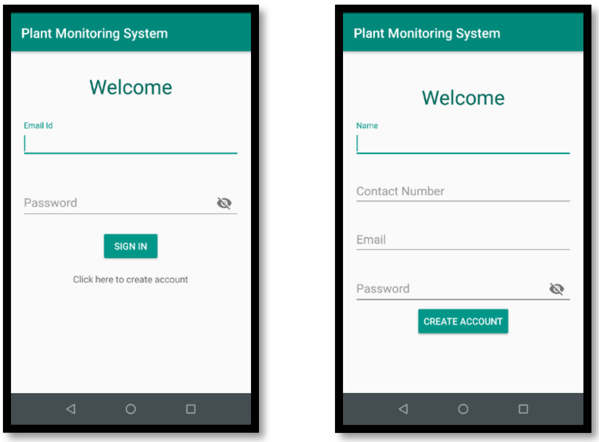


Fig. 5.2(a) 1) Sign-in screen 2) Create account screen

Once the user signs in, he can directly view all the data of his farm in the Application. If the user has just created a new account, he is prompted to add a hardware unit in the system. The hardware unit can be configured into the system very easily by just entering the hardware unit ID and the description. Once the hardware unit is configured into the system, the user just has to place the unit at the required location on the farm.

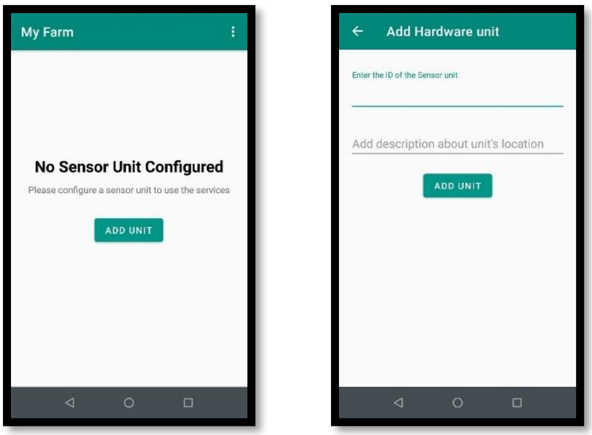


Fig. 5.2(b) 1) Screen prompting the user to add sensor unit 2) Add hardware unit screen

After placing the unit on the farm, the App shows the readings that the sensors have measured. But the user needs to wait for some time till the first reading is measured. During this phase, the Application does not show any data. It displays a message indicating the user that the device has been configured, but the hardware unit has not reported any data as of now.

Now when the system is up and running, if the user feels that one more unit should be added on the farm, it can be easily added from the Application. The user needs to go to the settings page of the Application. There are various options over there. One of the options allows the user to add a new hardware unit in the system. Clicking on the option opens the screen shown in Fig 5.2(b).

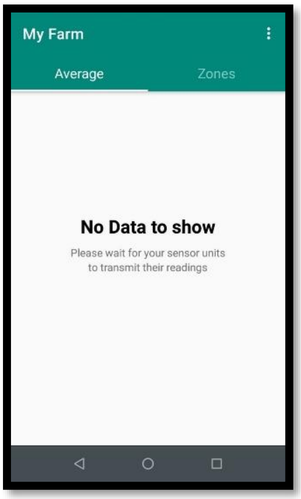


Fig. 5.2 (c) Screen showing that no readings are reported yet

The settings page enables various modifications in the system. It has one option to change the language of the Application. Users can change the language from English to one of his preferences. This Application supports Hindi, Gujarati and English as of now. There may be a situation when the user wants to change the location of the hardware unit. The location description of the hardware unit can be changed from the settings page. The settings page also shows the number of devices currently deployed in the farm. In the settings page, the user can also view all the units deployed in the farm along with their descriptions.

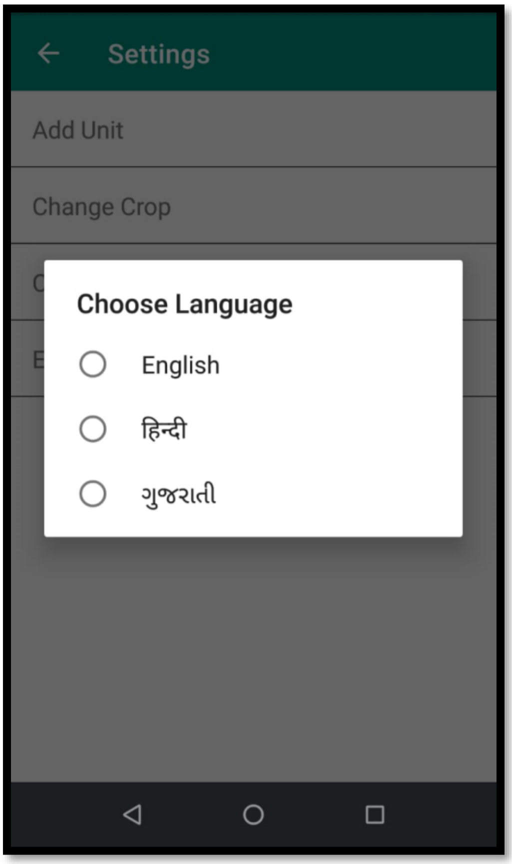


Fig. 5.2 (d) Screen prompting to choose language

On the main page of the App, we have an option from where we can share the data of the farm in PDF file format. Clicking on the “share farm data” option from the drop-down menu data can be shared. The user is now asked to select the platform where he would like to share the data. The data can be shared over various platforms such as WhatsApp, Email, or Google drive.

In the same drop-down menu, we have an option to go to the web page where the Web Application is hosted. After login into the Application, he can view the results produced by the Machine Learning Algorithms hosted on this web application.

Once the initial set up is completed, the user can now view all the data of the farm at the touch of a fingertip. As mentioned earlier, this data is updated in real-time without reloading or refreshing the Application. The image shown below shows a comparison between the data stored on the cloud and the data displayed in the Application.

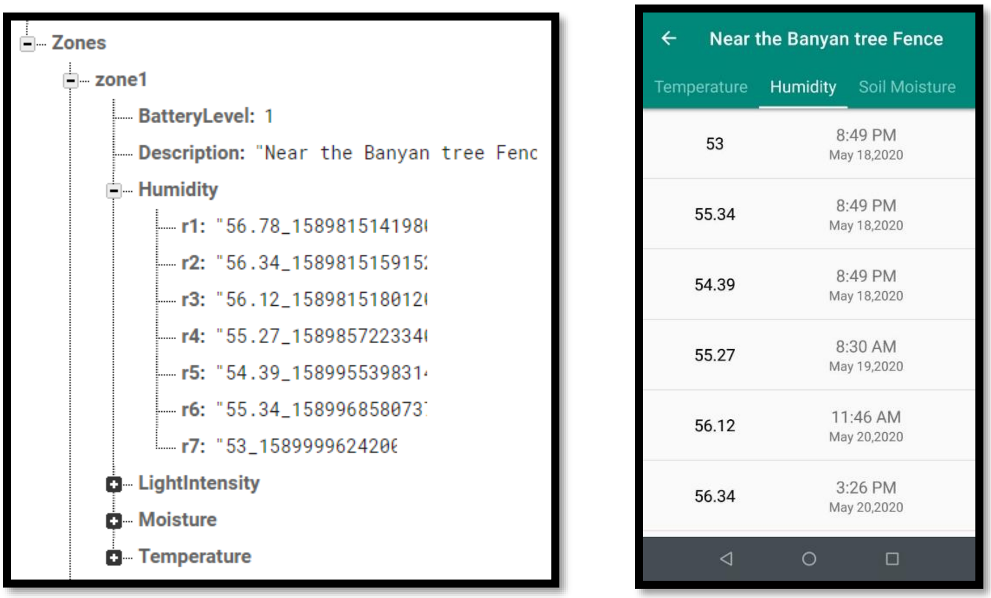


Fig. 5.2(e) Comparison between data stored on the cloud and displayed in the Application

Here, as we can see that the order in which the data is stored in the Firebase is different than what is displayed in the App. This is due to the fact that in Firebase, a new value is appended at the end of the list, but when we display the same in the App, we show the latest data on the top.

5.3 Machine learning predictions and results:

This section shows the results of the machine learning algorithm applied to the dataset. We have used linear and logistic regression to build the following models.

1. To predict whether it will rain or not based on the climatic conditions using logistic regression.
2. To determine the amount of rainfall using a multivariable linear regression technique.
3. To determine the amount of crop production based on the land area, season, and rainfall in the region using the classification and linear regression technique.

5.3.1 Rainfall prediction using logistic regression:

In the first section, we have used a local weather dataset to predict whether it will rain or not. The results are shown by preparing the assessing techniques of logistic regression, which involves confusion matrix, AUC, and accuracy check based on probability. The stepwise implementation of the model is given as:

1. To load the weather data set, which contains 22 columns of different parameters.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	MinTemp	MaxTemp	Rainfall	Evaporati	Sunshine	WindGust	WindGust	WindDir	WindDir	WindSpeed	WindSpeed	Humidity9am	Humidity3pm	Pressure9	Pressure3	Cloud9am	Cloud3pm	Temp9am	Temp3pm	RainToday	RISK_MM
2	8	24.3	0	3.4	6.3	NW	30	SW	NW	6	20	68	29	1019.7	1015	7	7	14.4	23.6	No	3.6
3	14	26.9	3.6	4.4	9.7	ENE	39	E	W	4	17	80	36	1012.4	1008.4	5	3	17.5	25.7	Yes	3.6
4	13.7	23.4	3.6	5.8	3.3	NW	85	N	NNE	6	6	82	69	1009.5	1007.2	8	7	15.4	20.2	Yes	39.8
5	13.3	15.5	39.8	7.2	9.1	NW	54	WNW	W	30	24	62	56	1005.5	1007	2	7	13.5	14.1	Yes	2.8
6	7.6	16.1	2.8	5.6	10.6	SSE	50	SSE	ESE	20	28	68	49	1018.3	1018.5	7	7	11.1	15.4	Yes	0
7	6.2	16.9	0	5.8	8.2	SE	44	SE	E	20	24	70	57	1023.8	1021.7	7	5	10.9	14.8	No	0.2
8	6.1	18.2	0.2	4.2	8.4	SE	43	SE	ESE	19	26	63	47	1024.6	1022.2	4	6	12.4	17.3	No	0
9	8.3	17	0	5.6	4.6	E	41	SE	E	11	24	65	57	1026.2	1024.2	6	7	12.1	15.5	No	0
10	8.8	19.5	0	4	4.1	S	48	E	ENE	19	17	70	48	1026.1	1022.7	7	7	14.1	18.9	No	16.2
11	8.4	22.8	16.2	5.4	7.7	E	31	S	ESE	7	6	82	32	1024.1	1020.7	7	1	13.3	21.7	Yes	0
12	9.1	25.2	0	4.2	11.9	N	30	SE	NW	6	9	74	34	1024.4	1021.1	1	2	14.6	24	No	0.2
13	8.5	27.3	0.2	7.2	12.5	E	41	E	NW	2	15	54	35	1023.8	1019.9	0	3	16.8	26	No	0
14	10.1	27.9	0	7.2	13	WNW	30	S	NW	6	7	62	29	1022	1017.1	0	1	17	27.1	No	0
15	12.1	30.9	0	6.2	12.4	NW	44	WNW	W	7	20	67	20	1017.3	1013.1	1	4	19.7	30.7	No	0
16	10.1	31.2	0	8.8	13.1	NW	41	S	W	6	20	45	16	1018.2	1013.7	0	1	18.7	30.4	No	0
17	12.4	32.1	0	8.4	11.1	E	46	SE	WSW	7	9	70	22	1017.9	1012.8	0	3	19.1	30.7	No	0
18	13.8	31.2	0	7.2	8.4	ESE	44	WSW	W	6	19	72	23	1014.4	1009.8	7	6	20.2	29.8	No	1.2
19	11.7	30	1.2	7.2	10.1	S	52	SW	NE	6	11	59	26	1016.4	1013	1	5	20.1	28.6	Yes	0.6
20	12.4	32.3	0.6	7.4	13	E	39	NNE	W	4	17	60	25	1017.1	1013.3	1	3	20.2	31.2	No	0
21	15.6	33.4	0	8	10.4	NE	33	NNW	NNW	2	13	61	27	1018.5	1013.7	0	1	22.8	32	No	0
22	15.3	33.4	0	8.8	9.5	WNW	59	N	NW	2	31	60	26	1012.4	1006.5	1	5	22.2	32.8	No	0.4
23	16.4	19.4	0.4	9.2	0	E	26	ENE	E	6	11	88	72	1010.7	1008.9	8	8	16.5	18.3	No	25.8
24	12.8	18.5	25.8	2.8	0.6	ESE	28	S	SE	13	13	91	79	1014	1014.9	8	8	14	16.8	Yes	0.4
25	12	24.3	0.4	1.2	7.5	NNE	26	WSW	NE	6	9	74	57	1020.7	1019.2	7	5	17.8	22.8	No	0
26	15.4	28.4	0	4.4	8.1	ENE	33	SSE	NE	9	15	85	31	1022.4	1018.6	8	2	16.8	27.3	No	0
27	15.6	26.9	0	6.8	8.9	E	41	E	E	6	22	65	48	1019.7	1016.5	2	4	19.8	25.1	No	0.2
28	13.3	22.2	0.2	6.6	2.3	ENE	39	E	E	20	17	70	55	1021	1018.6	7	7	16.5	21.2	No	0
29	12.9	28	0	4.4	10.7	S	52	S	NNE	6	11	61	31	1019.2	1014.8	5	7	18.8	26.7	No	0

Fig 5.3.1(a) Input weather dataset

2. To clean the dataset using pandas' libraries to get the data that is needed for prediction. The dataset is cleaned and converted to 12 useful columns.

```
     MinTemp  MaxTemp  Rainfall  Evaporation  Humidity9am  Humidity3pm
0        8.0     24.3       0.0          3.4           68           29
1       14.0     26.9       3.6          4.4           80           36
2       13.7     23.4       3.6          5.8           82           69
3       13.3     15.5      39.8          7.2           62           56
4        7.6     16.1       2.8          5.6           68           49
..       ...      ...       ...          ...          ...          ...
361      9.0     30.7       0.0          7.6           38           15
362      7.1     28.4       0.0         11.6           45           22
363     12.5     19.9       0.0          8.4           63           47
364     12.5     26.9       0.0          5.0           69           39
365     12.3     30.2       0.0          6.0           43           13

     Pressure3pm  Cloud9am  Cloud3pm RainToday  AvgHum  Cloud_cover
0         1015.0         7         7        No     1.0          7.0
1         1008.4         5         3       Yes     1.0          4.0
2         1007.2         8         7       Yes     1.0          7.5
3         1007.0         2         7       Yes     1.0          4.5
4         1018.5         7         7       Yes     1.0          7.0
..           ...       ...       ...       ...     ...          ...
361       1010.8         1         3        No     0.0          2.0
362       1016.9         0         1        No     0.0          0.5
363       1022.8         3         2        No     1.0          2.5
364       1016.2         6         7        No     1.0          6.5
365       1009.2         1         1        No     0.0          1.0

[366 rows x 12 columns]
```

Fig 5.3.1(b) processed dataset

3. Visualization data based on other parameters like cloud cover and humidity %

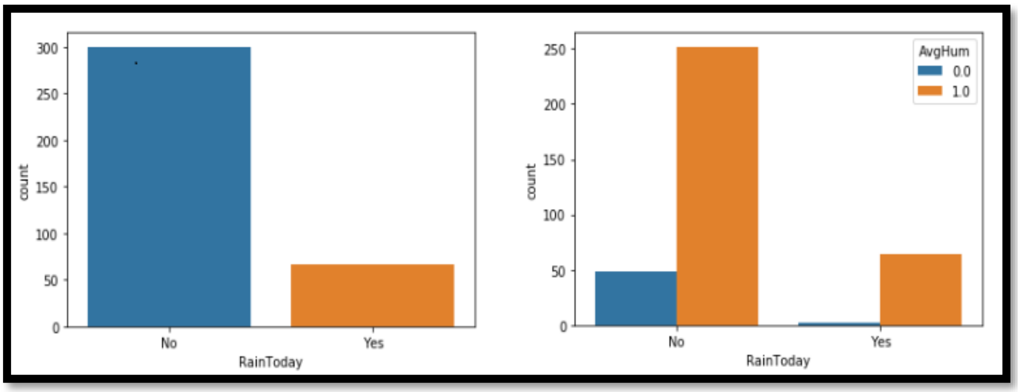


Fig 5.3.1(c) Count of rainy days classified on the basis of humidity

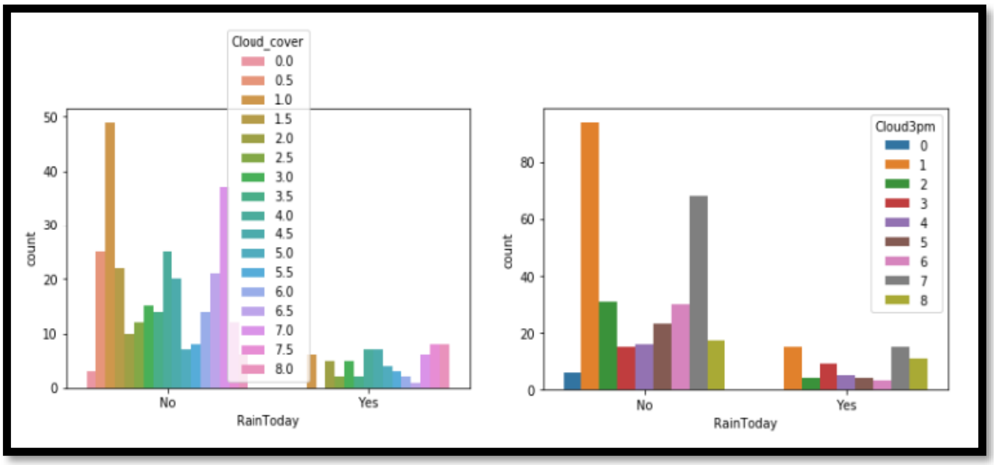


Fig 5.3.1(d) Count of rainy days classified based on cloud cover

4. To check the statistical parameters involved to get the insight of the data.

	MinTemp	MaxTemp	Rainfall	Evaporation	Humidity9am
count	366.000000	366.000000	366.000000	366.000000	366.000000
mean	7.265574	20.550273	1.428415	4.521858	72.035519
std	6.025800	6.690516	4.225800	2.669383	13.137058
min	-5.300000	7.600000	0.000000	0.200000	36.000000
25%	2.300000	15.025000	0.000000	2.200000	64.000000
50%	7.450000	19.650000	0.000000	4.200000	72.000000
75%	12.500000	25.500000	0.200000	6.400000	81.000000
max	20.900000	35.800000	39.800000	13.800000	99.000000

	Humidity3pm	Pressure3pm	Cloud9am	Cloud3pm	AvgHum
count	366.000000	366.000000	366.000000	366.000000	366.000000
mean	44.519126	1016.810383	3.890710	4.024590	0.863388
std	16.850947	6.469422	2.956131	2.666268	0.343907
min	13.000000	996.800000	0.000000	0.000000	0.000000
25%	32.250000	1012.800000	1.000000	1.000000	1.000000
50%	43.000000	1017.400000	3.500000	4.000000	1.000000
75%	55.000000	1021.475000	7.000000	7.000000	1.000000
max	96.000000	1033.200000	8.000000	8.000000	1.000000

Fig 5.3.1(e) statistical results of all the parameters

5. Applying the logistic regression model and predicting the results.

```
              precision    recall  f1-score   support

           0       0.98      1.00      0.99        87
           1       1.00      0.91      0.95        23

    accuracy                           0.98       110
   macro avg       0.99      0.96      0.97       110
weighted avg       0.98      0.98      0.98       110

[[87  0]
 [ 2 21]]
[0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1
 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 1 0 1 1 1 1 0 0 0]
```

Fig 5.3.1(f) Prediction and test results with confusion matrix

6. Creating a confusion matrix and determining probabilities to create the ROC curve.

```
from sklearn.metrics import accuracy_score
final_ans = accuracy_score(Y_test,prediction) * 100
print(final_ans)
```

```
98.18181818181819
```

Fig 5.3.1(g) Accuracy of the predicted values

```
0.5
1.0
No Skill: ROC AUC=0.500
Logistic: ROC AUC=1.000
```

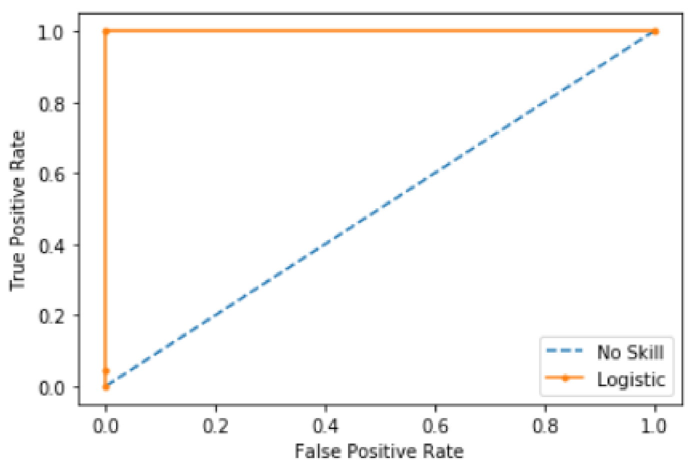
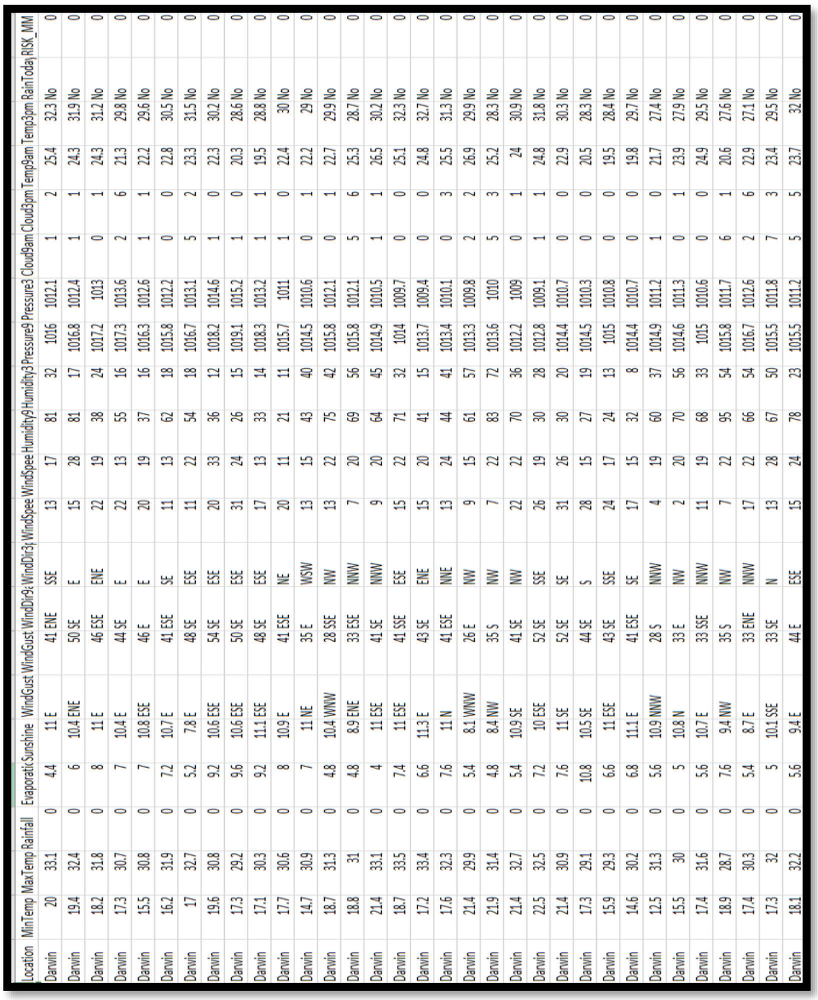


Fig 5.3.1(h) ROC Curve

5.3.2 Rainfall prediction using linear regression:

In the second section, we have used a dataset of weather conditions of Darwin to predict the amount of rainfall based on the surrounding weather conditions. The results are shown by preparing the assessing techniques of multivariable linear regression, which involves R^2 value and the value of error in actual and predicted results. The stepwise implementation of the model is given as:

1. To load the weather data set, which contains 24 columns of different parameters and 3011 rows of data.



Location	MinTemp	MaxTemp	Rainfall	Evaporatic	Sunshine	WindGust	WindGust	WindDir9a	WindDir3p	WindSpee	WindSpee	Humidity9	Humidity3	Pressure9	Pressure3	Cloud9am	Cloud3pm	Temp9am	Temp3pm	RainToday	RISK_MM
Darwin	20	33.1	0	4.4	11	E	41	ENE	SSE	13	17	81	32	1016	1012.1	1	2	25.4	32.3	No	0
Darwin	19.4	32.4	0	6	10.4	ENE	50	SE	E	15	28	81	17	1016.8	1012.4	1	1	24.3	31.9	No	0
Darwin	18.2	31.8	0	8	11	E	46	ESE	ENE	22	19	38	24	1017.2	1013	0	1	24.3	31.2	No	0
Darwin	17.3	30.7	0	7	10.4	E	44	SE	E	22	13	55	16	1017.3	1013.6	2	6	21.3	29.8	No	0
Darwin	15.5	30.8	0	7	10.8	ESE	46	E	E	20	19	37	16	1016.3	1012.6	1	1	22.2	29.6	No	0
Darwin	16.2	31.9	0	7.2	10.7	E	41	ESE	SE	11	13	62	18	1015.8	1012.2	0	0	22.8	30.5	No	0
Darwin	17	32.7	0	5.2	7.8	E	48	SE	ESE	11	22	54	18	1016.7	1013.1	5	2	23.3	31.5	No	0
Darwin	19.6	30.8	0	9.2	10.6	ESE	54	SE	ESE	20	33	36	12	1018.2	1014.6	1	0	22.3	30.2	No	0
Darwin	17.3	29.2	0	9.6	10.6	ESE	50	SE	ESE	31	24	26	15	1019.1	1015.2	1	0	20.3	28.6	No	0
Darwin	17.1	30.3	0	9.2	11.1	ESE	48	SE	ESE	17	13	33	14	1018.3	1013.2	1	1	19.5	28.8	No	0
Darwin	17.7	30.6	0	8	10.9	E	41	ESE	NE	20	11	21	11	1015.7	1011	1	0	22.4	30	No	0
Darwin	14.7	30.9	0	7	11	NE	35	E	WSW	13	15	43	40	1014.5	1010.6	0	1	22.2	29	No	0
Darwin	18.7	31.3	0	4.8	10.4	WNW	28	SSE	NW	13	22	75	42	1015.8	1012.1	0	1	22.7	29.9	No	0
Darwin	18.8	31	0	4.8	8.9	ENE	33	ESE	NNW	7	20	69	56	1015.8	1012.1	5	6	25.3	28.7	No	0
Darwin	21.4	33.1	0	4	11	ESE	41	SE	NNW	9	20	64	45	1014.9	1010.5	1	1	26.5	30.2	No	0
Darwin	18.7	33.5	0	7.4	11	ESE	41	SSE	ESE	15	22	71	32	1014	1009.7	0	0	25.1	32.3	No	0
Darwin	17.2	33.4	0	6.6	11.3	E	43	SE	ENE	15	20	41	15	1013.7	1009.4	0	0	24.8	32.7	No	0
Darwin	17.6	32.3	0	7.6	11	N	41	ESE	NNE	13	24	44	41	1013.4	1010.1	0	3	25.5	31.3	No	0
Darwin	21.4	29.9	0	5.4	8.1	WNW	26	E	NW	9	15	61	57	1013.3	1009.8	2	2	26.9	29.9	No	0
Darwin	21.9	31.4	0	4.8	8.4	NW	35	S	NW	7	22	83	72	1013.6	1010	5	3	25.2	28.3	No	0
Darwin	21.4	32.7	0	5.4	10.9	SE	41	SE	NW	22	22	70	36	1012.2	1009	0	1	24	30.9	No	0
Darwin	22.5	32.5	0	7.2	10	ESE	52	SE	SSE	26	19	30	28	1012.8	1009.1	1	1	24.8	31.8	No	0
Darwin	21.4	30.9	0	7.6	11	SE	52	SE	SE	31	26	30	20	1014.4	1010.7	0	0	22.9	30.3	No	0
Darwin	17.3	29.1	0	10.8	10.5	SE	44	SE	S	28	15	27	19	1014.5	1010.3	0	0	20.5	28.3	No	0
Darwin	15.9	29.3	0	6.6	11	ESE	43	SE	SSE	24	17	24	13	1015	1010.8	0	0	19.5	28.4	No	0
Darwin	14.6	30.2	0	6.8	11.1	E	41	ESE	SE	17	15	32	8	1014.4	1010.7	0	0	19.8	29.7	No	0
Darwin	12.5	31.3	0	5.6	10.9	NNW	28	S	NNW	4	19	60	37	1014.9	1011.2	1	0	21.7	27.4	No	0
Darwin	15.5	30	0	5	10.8	N	33	E	NW	2	20	70	56	1014.6	1011.3	0	1	23.9	27.9	No	0
Darwin	17.4	31.6	0	5.6	10.7	E	33	SSE	NNW	11	19	68	33	1015	1010.6	0	0	24.9	29.5	No	0
Darwin	18.9	28.7	0	7.6	9.4	NW	35	S	NW	7	22	95	54	1015.8	1011.7	6	1	20.6	27.6	No	0
Darwin	17.4	30.3	0	5.4	8.7	E	33	ENE	NNW	17	22	66	54	1016.7	1012.6	2	6	22.9	27.1	No	0
Darwin	17.3	32	0	5	10.1	SSE	33	SE	N	13	28	67	50	1015.5	1011.8	7	3	23.4	29.5	No	0
Darwin	18.1	32.2	0	5.6	9.4	E	44	E	ESE	15	24	78	23	1015.5	1011.2	5	5	23.7	32	No	0

Fig 5.3.2(a) Input Dataset of climate-based parameters

2. To clean the dataset using pandas' libraries to get the data that is needed for prediction. The information is cleaned to 3006 rows and seven columns.

```
      MinTemp  MaxTemp  Rainfall  Humidity9am  Humidity3pm  Pressure9am
0        20.0     33.1       0.0         81.0         32.0       1016.0
1        19.4     32.4       0.0         81.0         17.0       1016.8
2        18.2     31.8       0.0         38.0         24.0       1017.2
3        17.3     30.7       0.0         55.0         16.0       1017.3
4        15.5     30.8       0.0         37.0         16.0       1016.3
...       ...      ...       ...          ...          ...          ...
3006     23.8     31.1      85.0         95.0         71.0       1007.0
3007     23.8     27.5      24.4         96.0         94.0       1008.0
3008     23.3     33.5      37.2         86.0         60.0       1007.9
3009     26.2     33.2       0.0         79.0         71.0       1008.1
3010     25.6     33.2       0.6         81.0         66.0       1008.5

      Pressure3pm
0          1012.1
1          1012.4
2          1013.0
3          1013.6
4          1012.6
...           ...
3006       1003.9
3007       1005.8
3008       1004.6
3009       1005.6
3010       1004.7

[3006 rows x 7 columns]
```

Fig 5.3.2(b) Processed and cleaned dataset

3. Visualization data based on other parameters by scatter plots and heat maps of the dataset.

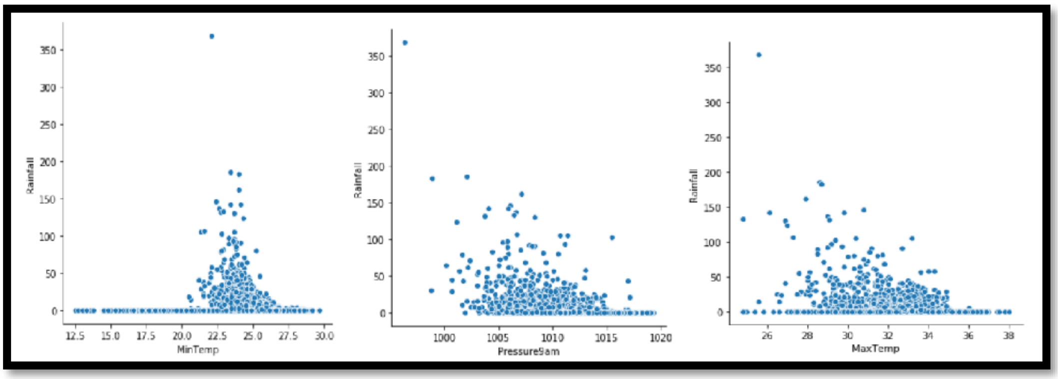


Fig 5.2.2(c) Scatter plot of temperature and pressure v/s rainfall

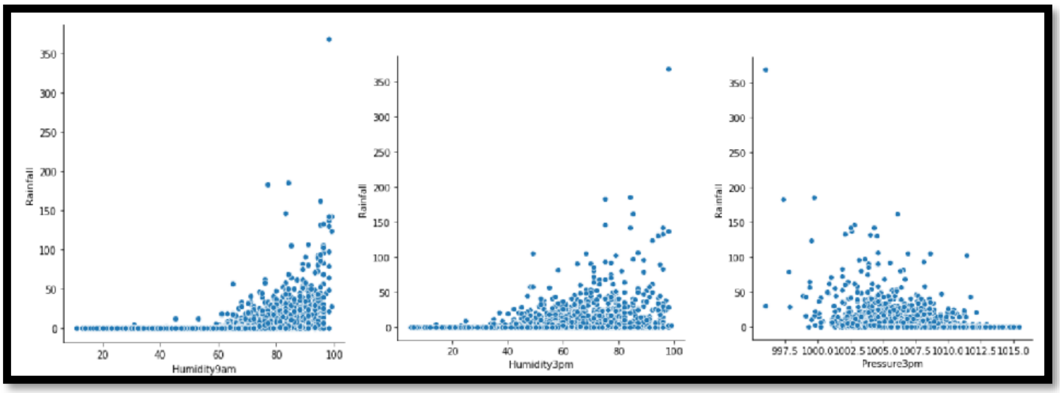


Fig 5.2.2(d) Scatter plot of temperature, pressure and humidity v/s rainfall

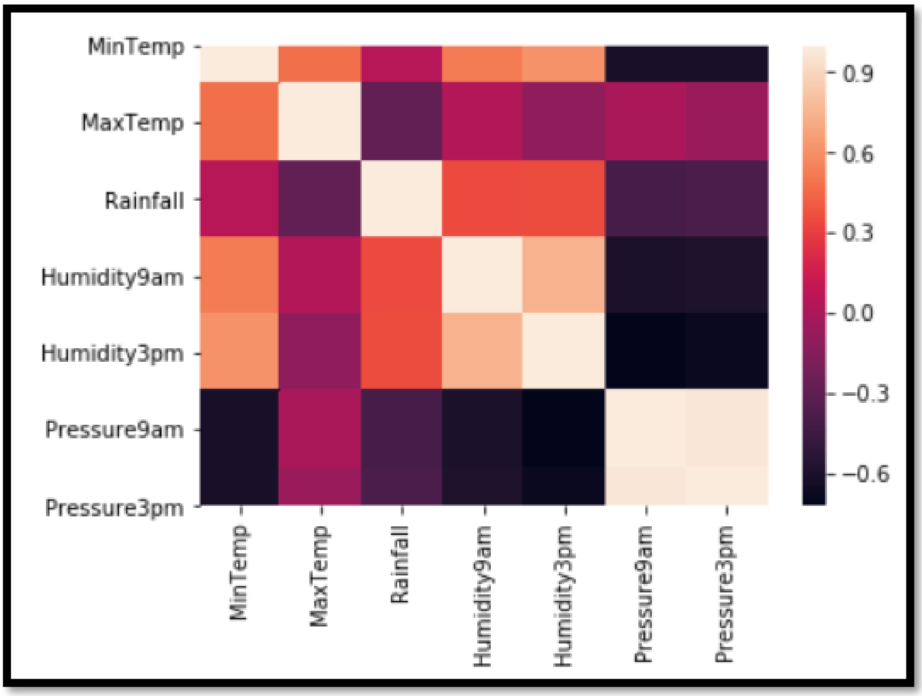


Fig 5.3.2(e) Heat map of the correlation of all the parameters

4. To check the statistical parameters involved to get the insight of the data.

	MinTemp	MaxTemp	Rainfall	Humidity9am	Humidity3pm
count	3006.000000	3006.000000	3006.000000	3006.000000	3006.000000
mean	23.174052	32.575516	4.768929	68.240852	51.520625
std	2.995185	1.740604	15.802909	14.964646	18.563111
min	12.500000	24.800000	0.000000	11.000000	5.000000
25%	21.500000	31.600000	0.000000	62.000000	39.000000
50%	23.900000	32.800000	0.000000	70.000000	52.000000
75%	25.300000	33.800000	1.400000	78.000000	63.000000
max	29.700000	38.000000	367.600000	99.000000	99.000000

	Pressure9am	Pressure3pm
count	3006.000000	3006.000000
mean	1011.972488	1008.402462
std	3.328501	2.887511
min	996.300000	996.000000
25%	1009.800000	1006.500000
50%	1012.300000	1008.700000
75%	1014.400000	1010.600000
max	1019.300000	1015.400000

Fig 5.3.2(f) Statistical parameters

5. Applying the linear regression model and predicting the results.

```
0 -1.249592
1 -1.995289
2 -0.529324
3  0.231030
4  0.035879
5 -1.655130
```

Fig 5.3.2(g) Regression coefficients

```
     Actual  Predicted
0       2.4  10.672129
1       0.0   8.808610
2       0.0   3.108484
3       0.0  -2.301406
4       0.0  -7.749481
..      ...        ...
597     1.0   8.353353
598    11.8   3.502904
599     1.6  17.954332
600     0.0  -9.411952
601     7.2  19.287737

[602 rows x 2 columns]
```

Fig 5.3.2(h) Vector of actual and predicted values

6. Creating a histogram between actual and predicted values.

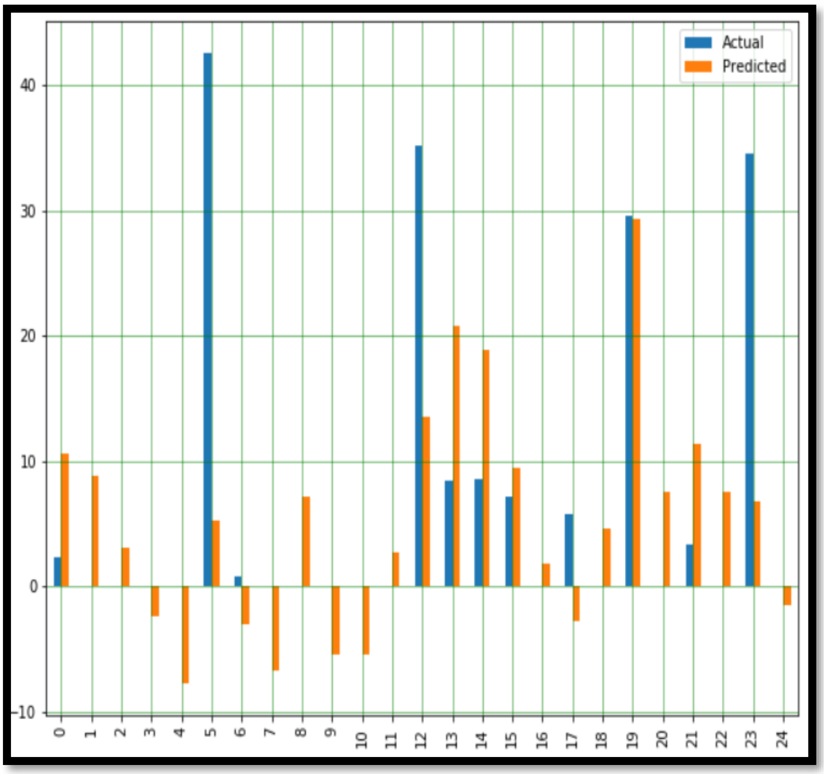


Fig 5.3.2(i) Histogram representation of actual v/s predicted values

5.3.3 Crop production prediction using linear regression:

In the third section, we have used a dataset of crops to predict the production of rabi and Kharif crop rice and wheat, respectively. The results are shown by preparing the assessing techniques of multivariable linear regression, which involves R^2 value and the value of error in actual and predicted results. The stepwise implementation of the model is given as:

1. To load the weather data set, which contains seven columns of different parameters and 74975 rows of data.

	A	B	C	D	E	F	G	H
1	Index	State	Year	Season	Crop	Area	Productior	Rainfall
2	0	Andaman a	2000	Kharif	Arecanut	1254	2000	2763.2
3	1	Andaman a	2000	Kharif	Other Khai	2	1	2763.2
4	2	Andaman a	2000	Kharif	Rice	102	321	2763.2
5	3	Andaman a	2000	Whole Yea	Banana	176	641	2763.2
6	4	Andaman a	2000	Whole Yea	Cashewnu	720	165	2763.2
7	5	Andaman a	2000	Whole Yea	Coconut	18168	65100000	2763.2
8	6	Andaman a	2000	Whole Yea	Dry ginger	36	100	2763.2
9	7	Andaman a	2000	Whole Yea	Sugarcane	1	2	2763.2
10	8	Andaman a	2000	Whole Yea	Sweet pot	5	15	2763.2
11	9	Andaman a	2000	Whole Yea	Tapioca	40	169	2763.2
12	10	Andaman a	2001	Kharif	Arecanut	1254	2061	3080.9
13	11	Andaman a	2001	Kharif	Other Khai	2	1	3080.9
14	12	Andaman a	2001	Kharif	Rice	83	300	3080.9
15	13	Andaman a	2001	Whole Yea	Cashewnu	719	192	3080.9

Fig 5.3.3(a) Input Dataset of crop

2. To clean the dataset using pandas' libraries to get the data that is needed for prediction. The information is cleaned to 22715 rows.

	Crop	Season	Area	Production	Rainfall
0	Rice	Kharif	102.00	321.00	2763.2
1	Rice	Kharif	83.00	300.00	3080.9
2	Rice	Kharif	189.20	510.84	2620.2
3	Rice	Kharif	52.00	90.17	2355.9
4	Rice	Kharif	52.94	72.57	2460.1
5	Rice	Kharif	2.09	12.06	2954.7
6	Rice	Autumn	3.50	10.00	3157.1
7	Rice	Kharif	10779.00	31863.00	2763.2
8	Rice	Kharif	9718.00	27033.00	3080.9
9	Rice	Kharif	6854.30	18995.62	2404.7

(22714, 5)

Fig 5.3.3(b) Processed and cleaned data

3. Visualization data based on other parameters by scatter plots and heat maps of the dataset.

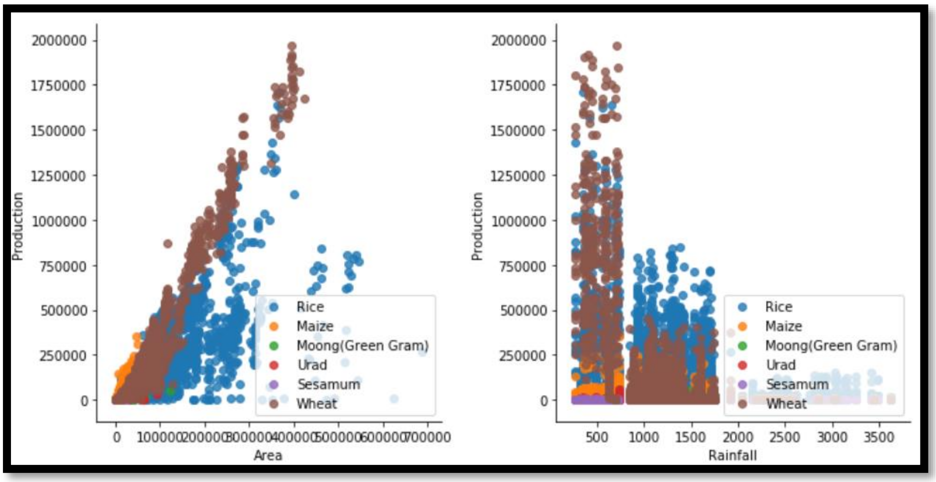


Fig 5.3.3(c) Scatter plot of production v/s area and rainfall classified with crops

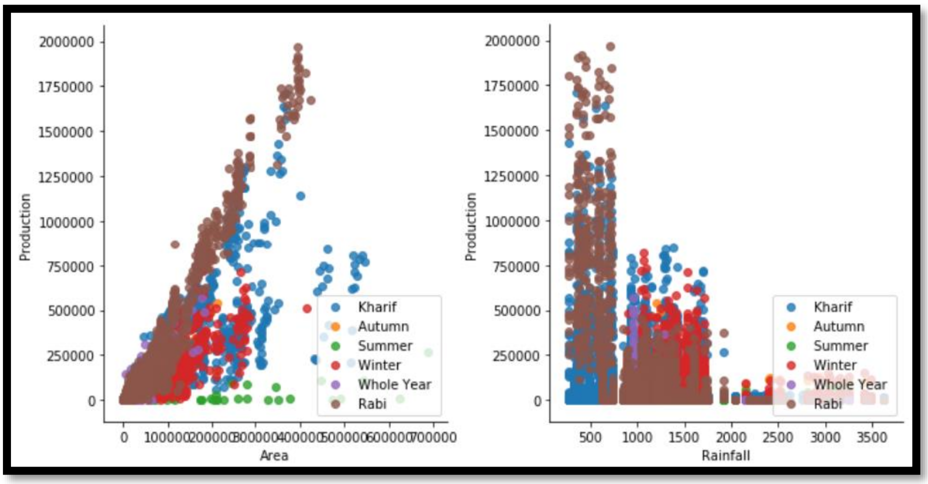


Fig 5.3.3(d) Scatter plot of production v/s area and rainfall classified with season

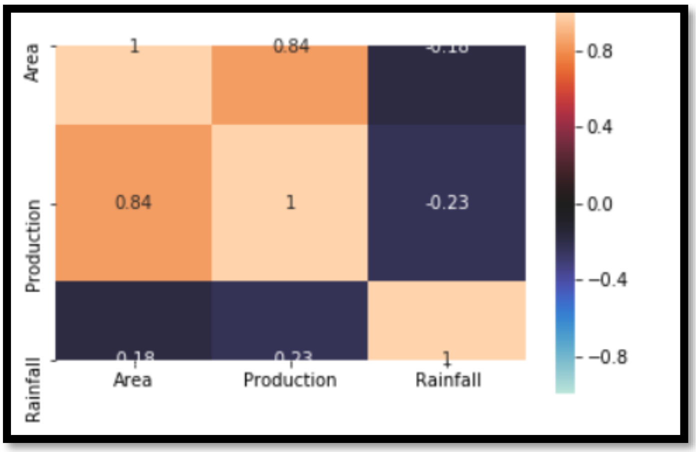


Fig 5.3.3(e) Heat map correlation of different parameters

4. To check the statistical parameters involved to get the insight of the data.

	Area	Production	Rainfall
count	22714.000000	2.271400e+04	22714.000000
mean	20757.648411	4.717063e+04	1297.327468
std	49828.552053	1.497862e+05	518.666966
min	0.280000	2.000000e-02	274.700000
25%	182.000000	1.004000e+02	1052.800000
50%	1806.000000	1.347000e+03	1278.700000
75%	13725.250000	1.923725e+04	1484.300000
max	687000.000000	1.969000e+06	3616.700000

Fig 5.3.3(f) Statistical parameters

5. Applying the linear regression model and predicting the results.

	0
0	1.882672
1	-30.456584

	Actual	Predicted
0	192.0	11017.045900
1	70708.0	148167.049666
2	295392.0	194194.827072
3	129942.0	185150.092247
4	887.0	21104.184897
...	...	...
1154	285510.0	338226.951525
1155	152083.0	157910.448696
1156	9263.0	34757.977585
1157	6210.0	-34105.033355
1158	70.0	17234.788969

[1159 rows x 2 columns]

Fig 5.3.3(g) Regression coefficients and Vector of actual and predicted values

6. Creating a histogram between actual and predicted values.

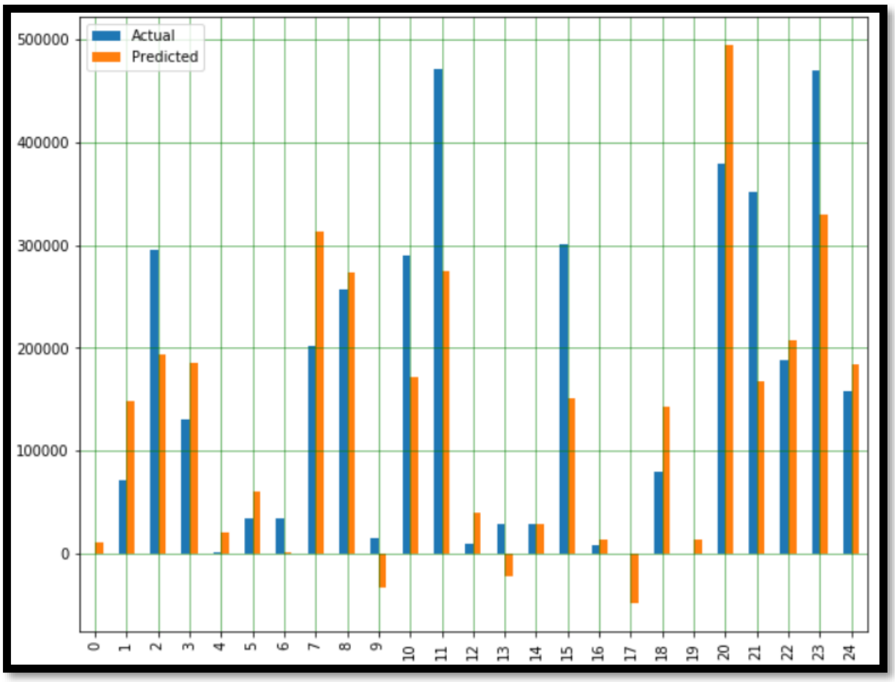


Fig 5.2.3(h) Histogram representation of actual v/s predicted values for rice

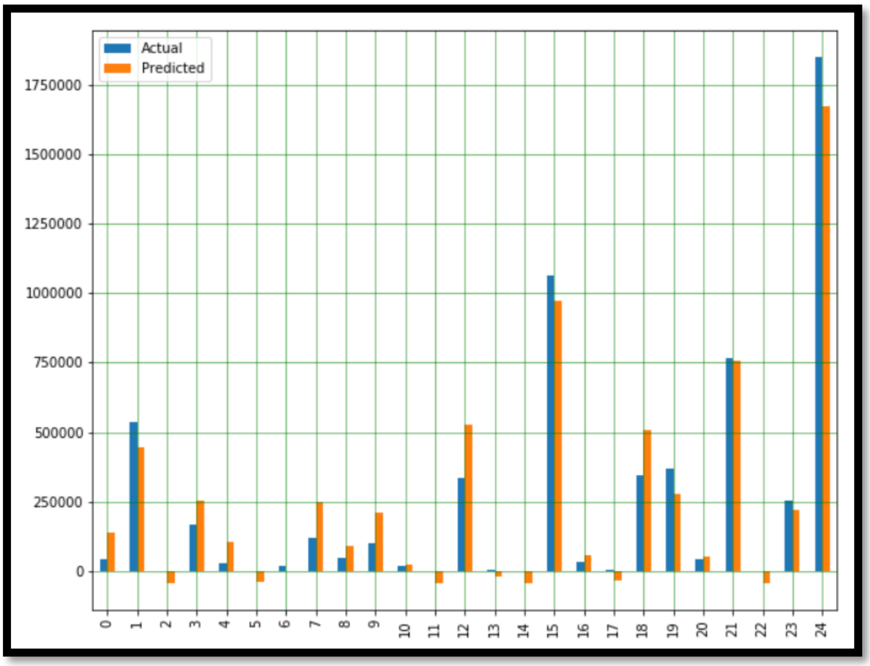


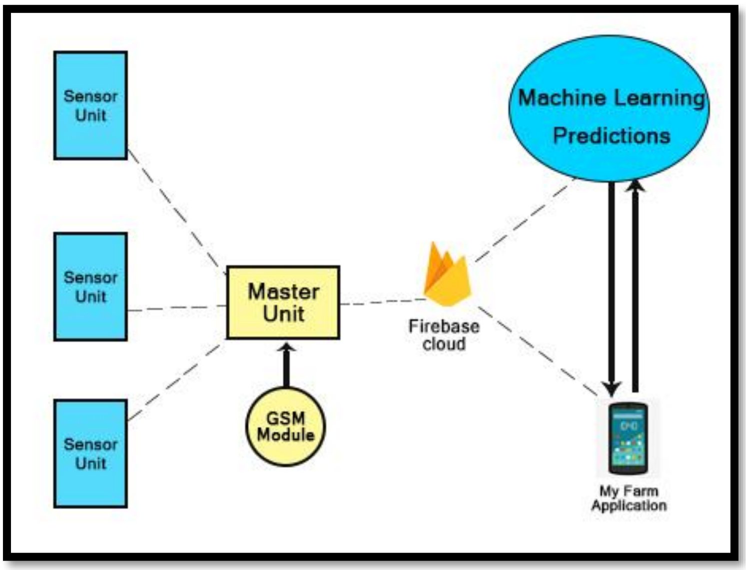
Fig 5.2.3(i) Histogram representation of actual v/s predicted values for rabi crops

****

CONCLUSION AND FUTURE SCOPE

In conclusion, we have implemented a system based on a combination of three domains, namely: IoT, Data Science, and machine learning. The system collects the data from the data acquisition system consisting of sensor units and NodeMCU. The collected data is stored on the firebase that can be used for displaying on mobile application and machine learning purposes. We have created a mobile app named "My Farm" to view the results and perform an important task on the database.

The next stage of this project would be to implement this system on a farm. Now, as farms are quite big, it is difficult and expensive to get Wi-Fi over the entire farm. So, rather than transmitting directly over Wi-Fi, different network architecture will be used. In this architecture, there will be many slave units and one master unit. The master unit alone will have access to the Internet via GPRS. Here the slave units consist of sensors for sensing, a processor for control and a device for wirelessly transmitting and receiving data (NRF24L01 module preferably). The slave units gather information about soil moisture, temperature and humidity. The information is transferred wirelessly to the master unit. From the master unit, the data is uploaded to the server, which can be accessed using an application on mobile phones. This information can be accessed from anywhere.



Once a one master one slave unit system is built, tested and verified, then multiple slave units can be connected wirelessly to a single master. Here, we can use the Polling method for master slave communication. Master unit will poll the slave unit for data. We can add external circuitry to find if the systems are functioning correctly or not. In future, the number of parameters measured by the system can also be increased. We can measure parameters such as soil temperature, the PH level of soil and nutrient level of the soil.

A web application can also be created for viewing the data. This web application will host the machine learning model which we have created. Thus, users will get to access the machine learning prediction results we have obtained. Moreover, the current notification can be improved to set crop wise threshold values. Chat forum over which farmers can share their knowledge or get their queries solved. A special system which detects the presence of any trespassing animal and alerts the user can also be incorporated. Thus, many more such modifications can be made to make the life farmers much easier.

****

REFERENCES

[1] "arduino IOT cloud," arduino, 2019. [Online]. Available: https://www.arduino.cc/en/IoT/HomePage.

[2] "developers," Google Developers, 2019. [Online]. Available: https://developer.android.com/studio/build.

[3] "Django documentaion," Open source framework, [Online]. Available: www.docs.djangoproject.com/en3.0.

[4] "Firebase documentation," Google Inc, [Online]. Available: www.firebase.google.com/docs. [Accessed Jan to May 2019-2020].

[5] "pandas for python," numFOCUS, 2016. [Online]. Available: https://pandas.pydata.org/. [Accessed 2020].

[6] *Protus design suit*, 2019.

[7] "Scikit-learn for machine learning," [Online]. Available: https://scikit-learn.org/stable/.

[8] M. S. S. S. Amit kumar agrawal, "Forecasting using machine learning," *International Journal technology and engineering* , 2019.

[9] B. T. Amruta Prabhakar Dhanfule, "Rainfall prediction using Linear regression," *International journal for scientic research and development*, vol. 6, no. 12, 2019.

[10] M. D. Hamza Benyezza, "Smart irrigation based think speak and arduino," *Internation conference on applied smart system*, 2018.

[11] M. L. J.Refonaa, "Rainfall predcition using regression model," *International journal of recent technology and engineering* , 2019.

[12] P. T. R. Jignesh Patoliya, "Brief review on wireless ros master-slave communiation using embedded IoT device:NodeMCU," *IEEE*, vol. 6, no. 2, 2019.

[13] D. L. Mark Holmstrom, "Machine learning applied to weather forecasting," *Stanforn University*, 2016.

[14] G. s. D. K. Neha khanna, "Design and development of soil moisture sensor and response monitoring system," *Internation journal of science and technology*, vol. 4, no. 6, p. 6, 2014.

[15] U. M. P.Priya, "International journal of engineering sciences and research technology," *Predicting yield of crop using machine learning algorithm,* 2018.

[16] S. D. m. Rajalakhsmi.P, "IoT based crop field monitoring and irrigation automation," 2012.

[17] D. G. Shweta Saraf, "IoT based smart irrigation monitoring and controlling system," *IEEE.*

[18] S. v. Varalakshmi, "Automatic plant escalation monitoring system using IoT," *Internation conference on computing and communication technology,* 2019.

[19] H. Documentation, "Hackerearth," Hackerearth, [Online]. Available: https://www.hackerearth.com/practice/machine-learning/machine-learning-algorithms/logistic-regression-analysis-r/tutorial/. [Accessed 2020].

[20] "towardsdatascience," [Online]. Available: https://towardsdatascience.com/.

ACRONYMS

ADC	Analog to digital converter
API	Application program interface
AUC	Area under curve
BSS	Basic service set
CdS	Cadmium supplied
DCF	Distributed control function
EN	Enable
GPIO	General purpose input output
I2C	Inter IC bus
I2S	Inter IC sound
IoT	Internet of things
IP	Internet protocol
IR	Infrared
LDR	Light dependent resistor
MAC	Media access control
MCU	Microcontroller unit
ML	Machine learning
NRF	Nordic Radio frequency
NTC	Negative temperature coefficient
P2P	Point to Point
PCB	Printed circuit board
PTC	Positive temperature coefficient
PWM	Pulse width modulation
RAM	Random access memory
RFID	Radio frequency identification
ROM	Read only memory
R-PI	Raspberry pi
RST	Reset
RTOS	Real time operating system
RX	Receiver
SPI	Serial peripheral interface
TCP	Transmission control protocol
TX	Transmitter
UART	Universal asynchronous receiver
UI	User interface
USB	Universal serial bus
WLAN	Wide area local area network
CSV	Comma-separated values
CPU	Central processing unit
FCM	Firebase cloud messaging
FNR	False negative rate
FPR	False positive rate
GCM	Google cloud messaging
IDE	Integrated development environment

HTTPS	Hypertext transfer protocol secure
MAE	Mean absolute error
MCU	Microcontroller unit
RMSE	Root mean squared error
ROC	Receiver operator characteristics

****