A-IoT CUBE: Smart Farm Soil Monitoring using IoT

Project report submitted to
Visvesvaraya National Institute of Technology,
Nagpur in partial fulfillment of the requirements
for the award of the degree

Bachelor of Technology
In
Mechanical Engineering
by

Animesh Jha	**(BT17MEC012)**
Dhananshu Gupta	**(BT17MEC031)**
D.Vivek Vardhan	**(BT17MEC034)**
Harshal Wadekar	**(BT17MEC040)**

under the guidance of

Dr. SHITAL CHIDDARWAR

Department of Mechanical Engineering
Visvesvaraya National Institute of Technology
Nagpur 440 010 (India)

April 2021

A-IoT CUBE: Smart Farm Soil Monitoring using IoT

Project report submitted to
Visvesvaraya National Institute of Technology,
Nagpur in partial fulfillment of the requirements
for the award of the degree

Bachelor of Technology
In
Mechanical Engineering
by

Animesh Jha (BT17MEC012)
Dhananshu Gupta (BT17MEC031)
D.Vivek Vardhan (BT17MEC034)
Harshal Wadekar (BT17MEC040)

under the guidance of

Dr. SHITAL CHIDDARWAR

Department of Mechanical Engineering
Visvesvaraya National Institute of Technology
Nagpur 440 010 (India)

April 2021

Department of Mechanical Engineering

Visvesvaraya National Institute of Technology, Nagpur

<u>Declaration</u>

We, Animesh Jha, Dhananshu Gupta, D. Vivek Vardhan and Harshal Wadekar, hereby declare that this project work titled "**A-IoT CUBE – Smart Farm Soil Monitoring using IoT**" is carried out by us in the Department of Mechanical Engineering of Visvesvaraya National Institute of Technology, Nagpur. The work is original and has not been submitted earlier whole or in part for the award of any degree/diploma at this or any other Institution / University.

Sr.No.	**Enrollment No**	**Names**	**Signature**
1	BT17MEC012	Animesh Jha	
2	BT17MEC031	Dhananshu Gupta	
3	BT17MEC034	D.Vivek Vardhan	D.Vivek Vardhan
4	BT17MEC040	Harshal Wadekar	Harshal

Date:

Department of Mechanical Engineering

Visvesvaraya National Institute of Technology, Nagpur

Certificate

This is to certify that the project titled "**A-IoT CUBE- Smart Farm Soil Monitoring using IoT**", submitted by **Animesh Jha, Dhananshu Gupta, D.Vivek Vardhan, Harshal Wadekar** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology in Mechanical Engineering**, VNIT Nagpur. The work is comprehensive, complete, and fit for final evaluation.

Dr. SHITAL CHIDDARWAR

Associate Professor

Department of Mechanical Engineering,

VNIT, Nagpur

Dr. S. B. THOMBRE

Head, Department of Mechanical Engineering

VNIT, Nagpur

Date:

Acknowledgement

We take this opportunity to acknowledge with a deep sense of gratitude to our project guide Dr. SHITAL CHIDDARWAR, Associate Professor, Department of Mechanical Engineering, VNIT Nagpur, for her impeccable guidance. We are also thankful to her for her constant motivation and encouragement that she provided to us throughout the project and her innovative solutions during the tenure. We are also deeply indebted to her for all her able guidance and moral support during the project. We have a deep sense of thankfulness for the valuable input provided by the Panel guides, at each evaluation step. We also thank Dr. S.B.Thombre, Head of Department of Mechanical Engineering, VNIT Nagpur, for his support.

Abstract

The Internet of Things (IoT) has tremendous success in health care, smart cities, industrial production, and so on. The Internet of Things (IoT) will become the future of every field transforming everyone's life by making everything intelligently. It is a network of variegated devices that make a self-configuring network. The new developments of Smart Farming with use of IoT, by day, turning the squatter of conventional Agricultural methods by not only making it optimal but moreover making it forfeit efficient for farmers and reducing yield wastage. The aim is to propose a technology that can generate messages on variegated platforms to notify farmers in order to take necessary decision. The product will squire farmers by getting live of (Temperature, humidity, soil moisture, ph. value) from the farmland in order to take necessary steps to which enable them in increasing their production and saving resources (water, fertilizers). The Microcontroller to be used is ARDUINO UNO / NODEMCU to connect all Sensors, 3D Printed cubes to house the apparatus and to be deployed in the farm, GSM Module for ARDUINO allows an Arduino board to connect to the internet, send and receive SMS, and make voice calls using the GSM library, ESP Module to be used to transfer data to cloud through WIFI, Jumper Wires Male to Male, male to female, female to female, Temperature and Humidity Sensor, Soil Moisture Sensor, Soil pH Value sensor and some resistances, Low energy wide area networking devices. This will indulge farmers to manage their yield with the new age in farming.

LIST OF FIGURES

LIST OF TABLES

LIST OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Introduction

India, having a population of 135 crores and Agriculture being 18% of the GDP, employing 59% of the population, it's undoubtedly one of the biggest occupational sectors. Unfortunately, only a limited portion of the earth's surface is suitable for threshing uses due to various limitations, like temperature, climate, topography, and soil quality, and even most of the suitable areas are not homogenous. When zooming the versatility of landscapes and plant types, many new differences start to Sally that can be difficult to quantify. Moreover, the agricultural land is remotely shaped by political and economic factors, like land and climate patterns and population density, while rapid urbanization is constantly posing threats to the availability of arable land.

Over the past decades, the total agricultural land utilized for supplies production has experienced a ripen. In 1991, the total arable area for supplies production was 19.5 million square miles (39.47% of the world's land area), which was reduced to approximately 18.6 million square miles (37.73% of the world's land area) in 2013. As such, the gap between the demand and supply of supplies is increasingly significant and viperous with time. Farming in India is washed-up using routine ways. The fact that most of our farmers lack proper knowledge makes this plane increasingly unreliable. A large portion of farming and agricultural activities are based on the forecasts, which at times fail. Farmers have to withstand huge losses and many times they end up committing suicide. Since we know the benefits of proper soil moisture and its quality, air quality and irrigation, in the growth of crops, such parameters cannot be ignored. Measuring these parameters and interpreting them in the simplest way using low-cost A-IoT devices is a crucial problem.

The future of Smart Computing will be completely based on the Internet of Things (IoT). It has a crucial role in transforming "Traditional Technology" from homes to offices to "Next Generation Everywhere Computing". 'Internet of Things' is gaining an important place in research wideness the world and expressly in the zone of wide wireless communications. Today IoT has started touching people everywhere and from the point of a normal user, IoT is laying the foundation for the minutiae of various products like smart health services, smart living, smart education in schools, and automation. And commercially it is the stuff used in manufacturing, transportation, agriculture, and merchant management, and many other fields. The most researched zone of IoT in agriculture. Because it is a crucial sector to ensure supply security as the global population is increasing rapidly. Researchers first started applying ICT-based techniques in this sector, which was useful on some levels but was not going to solve our problem in the long run. So now, they are exploring IoT as an option to ICT in agriculture. Agriculture products need applications like soil moisture monitoring, environmental condition monitoring for temperature, moisture, supply uniting management, and infrastructure management.

The future of Agriculture is precision Agriculture and it is expected to grow at 4 billion by 2020. Data generated from sensors in the Agriculture field can moreover be used for data analytics, which will help farmers to modernize yield yields. So, IoT based smart farming can solve many agriculture-based issues. This paper aims to introduce a working product that will indulge farmers with real-time data. Crops planted by farmers grow in nature, which is dynamic and unauthentic by unpredictable factors, such as weather and soil conditions, pest and disease challenges, and waffle yield conditions. These factors stupefy not only planting natures but moreover potential outcomes. Usually, soil samples are either checked manually or placid and sent to Agricultural testing Labs which is time-consuming, unreliable and reports are not in the local language. Smart farming using IoT is a combination of information technology and sustainable devices to increase efficiency, subtract managerial costs, and provide expert knowledge needed for farmers in agricultural management. IoT in Smart Farming Smart farming is a modern farming management concept with IoT technology to increase productivity in agriculture.

With the use of smart farming, farmers can powerfully use fertilizers and other resources to increase the quality and quantity of their crops. Farmers cannot be physically present in the field 24 hours a day. Also, the farmers may not have the knowledge to use variegated tools to measure the platonic environmental conditions for their crops. IoT provides them with a streamlined system that can function without any human supervision and can notify them to make proper visualization to deal with variegated kinds of problems they may squatter during farming. It has the sufficiency to reach and notify the farmer plane if the farmer is not on the field, which can indulge farmers to manage increasingly farmland, thus improving their production. IT is unscientific that the global population will reach the 9 billion mark by 2050. IoT use is a must for Agriculture to feed such a large population and powerfully use the farmland and other resources as they are scarcely misogynist in some places. Because of Global warming, unpredictable weather conditions are well-expressed the crops and farmers are facing major losses so the IoT Smart Farming using will indulge them to take quick measures to prevent that from happening. WSN which is moreover the workshop of IoT moreover includes the routing algorithms for a network like such increasing prototypes.

The benefits of Smart Farming are as follows: People are still working on variegated Smart Farming technology using IoT, so the predictable benefits of this technology are, Remote monitoring for farmers, water and other natural resource conservation, good management also allows improved livestock farming, the things which are not visible to the necked eye can be seen resulting is accurate farmland and yield evaluation, good quality as well as improved quantity, the facility to get the real-time data for useful insights. Shortfalls of Smart Farming are as follows

- Agricultural stuff relies mostly on nature, and man predicts or controls nature let it be rain drought sunlight availability. pest's tenancy etc. So overly implement IoT system agriculture.
- Smart threshing needs availability on the internet continuously. The rural part of the developing countries did not fulfil this requirement. Moreover, the internet is slower.
- The smart farming-based equipment requires a farmer to understand and learn the use of technology. This is the major Rencontre in raising smart threshing framing at a large-scale wideness the continues.
- Faulty sensors or data processing engines can rationalize faulty l decisions which may lead to overuse of water, fertilizers, and other wastage of resources.
- As data is transferred so the proper security of data is one of the shortfalls.

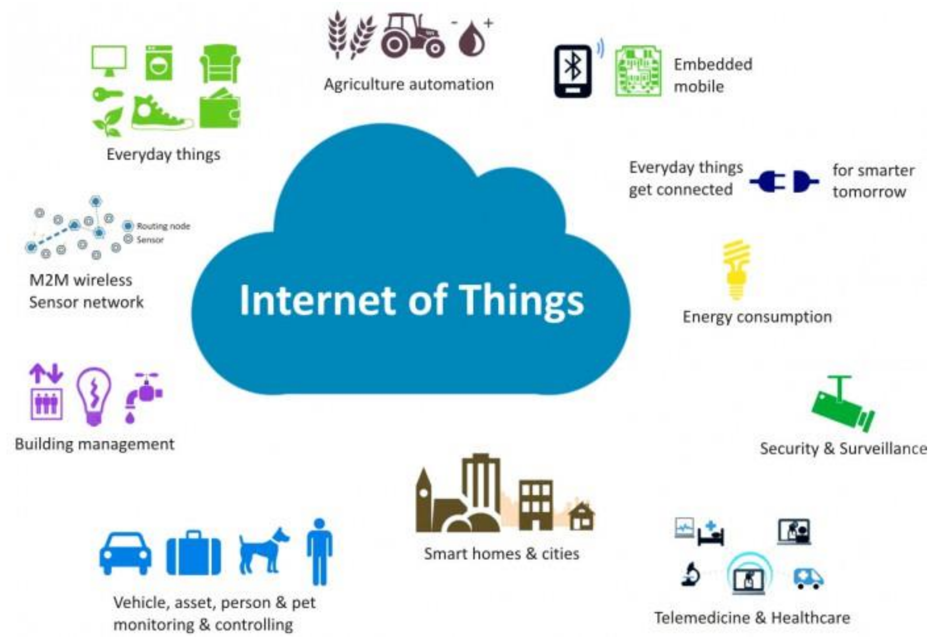

Figure 1 IoT Applications

1.2 Objective

The quick objective of the project goes as follows:

- The real-time data collection of various soil characteristics using sensors with the help of Arduino and processing this data to bring out useful insights.
- All the sensors are enabled and planted at their respective positions embedded in the IoT cube safely.
- Then they are connected with the Arduino board while the power supply will be from the solar panel attached above the top of the cube.
- Using Artificial Intelligence to portray and predict the trend of their farm characteristics in a simple and understanding way for the farmer.
- The manipulations on the data can be easily done using any AI software like Machine Learning etc.
- Alerting the farmer about the current condition of the soil through SMS alert / Mobile application with a user-friendly GUI.

- Help farmers to take decisions on what to grow next and what and when to add and how much to add.
- This has a greater impact on their daily as well as future decisions regarding the farm.

The figure below represents the plan of the project.

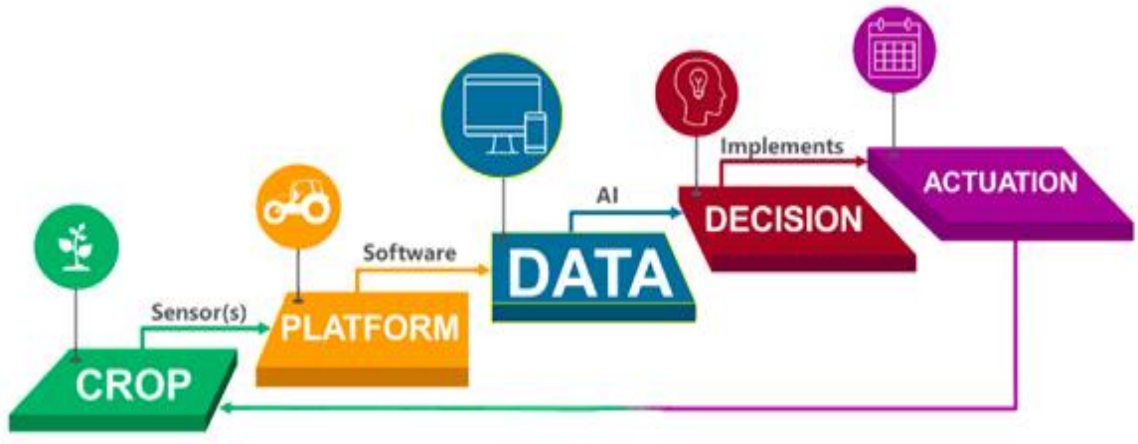

Figure 2 Plan of project

The cubes are then connected to the master cube which then transmits the information to the ‘ThingSpeak’ channel that has been created beforehand.

The final assembly is then installed in a cube which is powered by a solar cell using a solar panel on its top. The cube is then installed on a hollow tube acting as a pole for the system through which all the jumper wires are drawn out into the soil for the measurement. A number of cubes like this are installed at specific locations in the farm to complete the network of IoT cubes.

Overall, this project aims to interpret valuable information from farms to help improve the crop yield and optimize the use of resources. The figure depicts how the farm would be after all the IoT cubes are assembled and connected with the master cube.

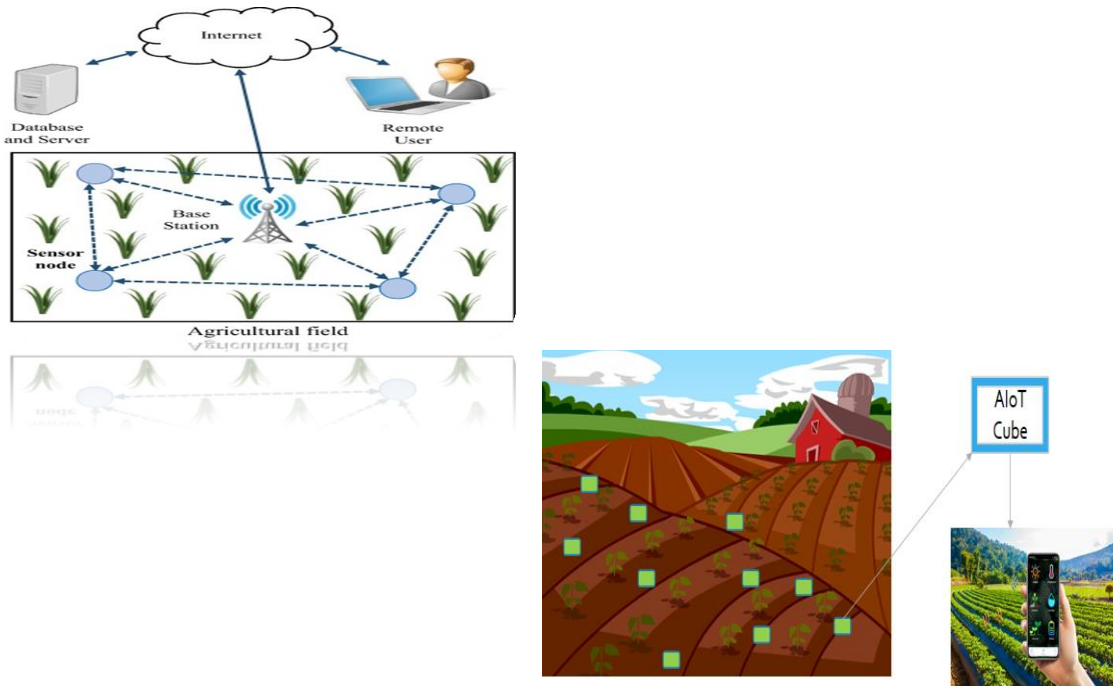

Figure 3 Pictorial representation of working of A-IoT Cube

CHAPTER 2

LITERATURE REVIEW

2.1 Internet-of-Things (IoT)- Based Smart Agriculture: Towards making the fields talk.

Authors: Muhammad Ayaz, Mohammad Ammad-Uddin, Zubair Sharif, Ali Mansour, El-Hadi M. Aggoune

Date of publication: August 1, 2019

Work Done:

This research paper is a review-based document providing insights of the State-of-the-art IoT based agriculture practices that are being carried out in the world, highlighting upon the scope of the inculcating IoT based technology in farming, it's application, potential soil parameters that are vital, identifying current and future trends in IoT agriculture. It also covers the concept of camera-based crop monitoring, which involves image processing for detection of anomalies, monitoring the plant health, use of UAVs in agriculture.

In today's era, the agriculture practices carried out are more precise than ever, more data-centric and smarter. This article highlights the potential of wireless sensors and IoT in agriculture, as well as the challenges expected to be faced when integrating this technology with the traditional farming practices.

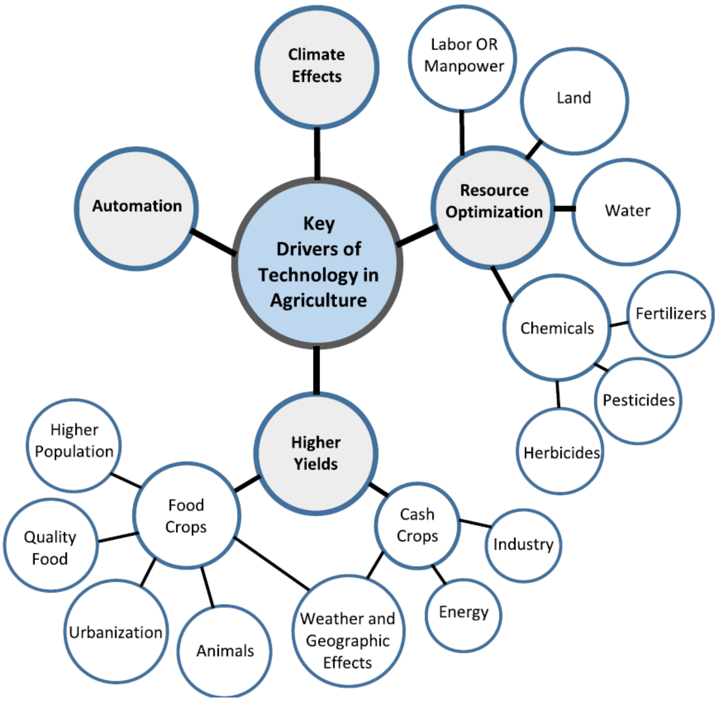

Figure 4 Key drivers in agricultural technology

By following the practices of smart agriculture, IoT can help to improve the solutions of many traditional farming issues, like drought response, yield optimization, land suitability, irrigation, and pest control. The current advanced methods of farming that include green-house farming, vertical farming, hydroponic, phenotyping, all these are heavily backed by the use of various sensors and real time data collection and interpretation, and thus stand upon the quality they consistently deliver.

The major equipment and technologies that are inculcated in the sector of agriculture are discussed below as per the paper:

A. Wireless Sensors:

Wireless sensors are being used standalone wherever required, further integrated with almost every portion of advanced agricultural tools and heavy machinery, depending on application requirements. This includes acoustic sensors, optical sensors, ultrasonic range sensors, electrochemical sensors, mechanical sensors, mass flow sensors, LIDAR sensors, etc. Each of these found heavy usage in soil monitoring, pest detection and control, fertilizer requirements, plant health monitoring, basically playing the part in first hand measurement of various parameters, which generates data to be further transferred and processed.

B. IoT Based Tractors:

As rural labour resources have started to come under stress due to the expansion of the crop industry, tractors and other automatic heavy machinery started to enter the agriculture sector. To fulfil the continuously increasing demands, agricultural-based equipment manufacturers, like John Deere, Hello Tractors, Case IH and CNH (New Holland), have started to provide better solutions focusing on the grower's requirements. With the advancement of technology, most of these manufacturers are offering tractors with automatic-driven and even Cloud-computing capabilities.

This technology is not new, as self-driving tractors have been in the market even before semi-autonomous cars. One of the main advantages of self-driving tractors is their ability to avoid revisiting the same area or row by reducing the overlap even less than an inch. In addition, they can make very precise turns without a driver's physical presence. This facility offers better precision with reduced errors, especially when spraying insecticide or targeting weeds; those are mostly unavoidable when a human control the machinery.

C. Harvesting Robots:

Harvesting is an essential phase during the production process, as it commands the crop's output and, ultimately, its success. In some crops, this is done a single time while, in some others, performed several times, even on a daily basis, as the crop reaches a certain stage. There’s always a right time to harvest the crop in order to ensure an optimized yield, any time other than it can affect the production notably. To automate this step, the role of robots has been significantly increasing in use, ex. In fruit detection, automatic crop harvesting, fruits plucking etc.

D. Communication in Agriculture:

Communication and reporting the information on a timely basis are a vital component in the field of precision farming. To achieve this, requires a reliable, and secure connection among constituents. To achieve communication reliability, telecom operators can play a crucial role in the agricultural sector.

Cellular communication largely depends on the bandwidth that is available at a particular location but is one of the most widely used tools for communication. However, the availability for network in rural areas is a concern and hence other modes for ex. LPWAN is used as a better solution.

Zigbee is a technology to communicate over relatively short-range communications, yet covering a wide range of applications. The devices based on this tool can be any one from Coordinator, Router or End User. Real time data is sent from the sensor nodes to the Zigbee protocol during monitoring.

LoRa is a wireless technology extensively used in IoT based applications, the benefits of it being its low power consumption, as it offers LPWAN connectivity between the sensor nodes and the Cloud server, it's more reliable and effective as compared to other available options (ex. WiFi and Bluetooth). The signals generated has the ability to penetrate through thick walls and insulated objects, thus can easily cover a larger area.

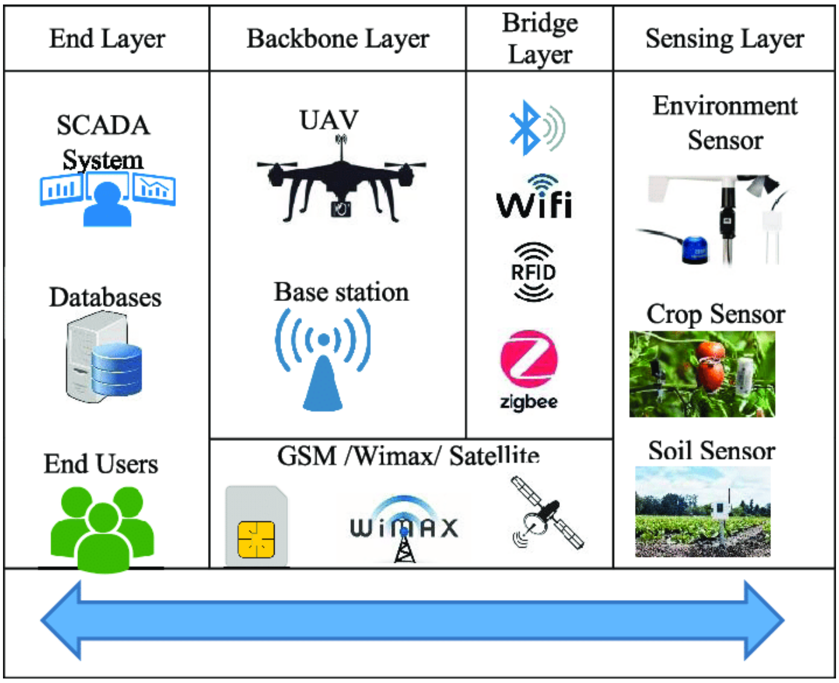

Figure 5 Communication Layers

E. Cloud Computing:

Precision in the field of agriculture calls for it's operations to use decision making that is data-driven. This data that is collected remotely has to be brought to a cloud server to further process it by applying specific algorithms and thus making feasible field decisions. Farmers can access the cloud server to get in the information that has been generated to choose the right product for their specific requirements.

Communication/ Data type	Possible application	Expected Data size	Power consumption (active mode)
Small sized data and low power consumption	(1) Air temperature/ humidity/ direction / speed (2) Soil temperature/ humidity (3) Leaf thickness/color (chlorophyll) (4) Trunk thickness/flux flow (5) Fruit size	100s of bytes	Less than a mA (Fractions of mA)
Medium sized data and medium power consumption	(1) Still picture camera (2) Multi or hyper spectral camera (3) Acoustic sensors	10s of Mb	10s of mA
Large sized data and large power consumption	Video streaming cameras	10s of Mb per minute	50 A

Table 1 Data and Power Consumption Chart

2.2. State-of-the-Art Internet of Things in Protected Agriculture

Authors: Xiaojie Shi, Xingshuang An, Qingxue Zhao, Huimin Liu,

Lianming Xia, Xia Sun, Yemin Guo

Year of publication: 2019

Work Done:

In this research, based on the actual situations of protected agriculture they have proposed a five – layer IoT architecture.

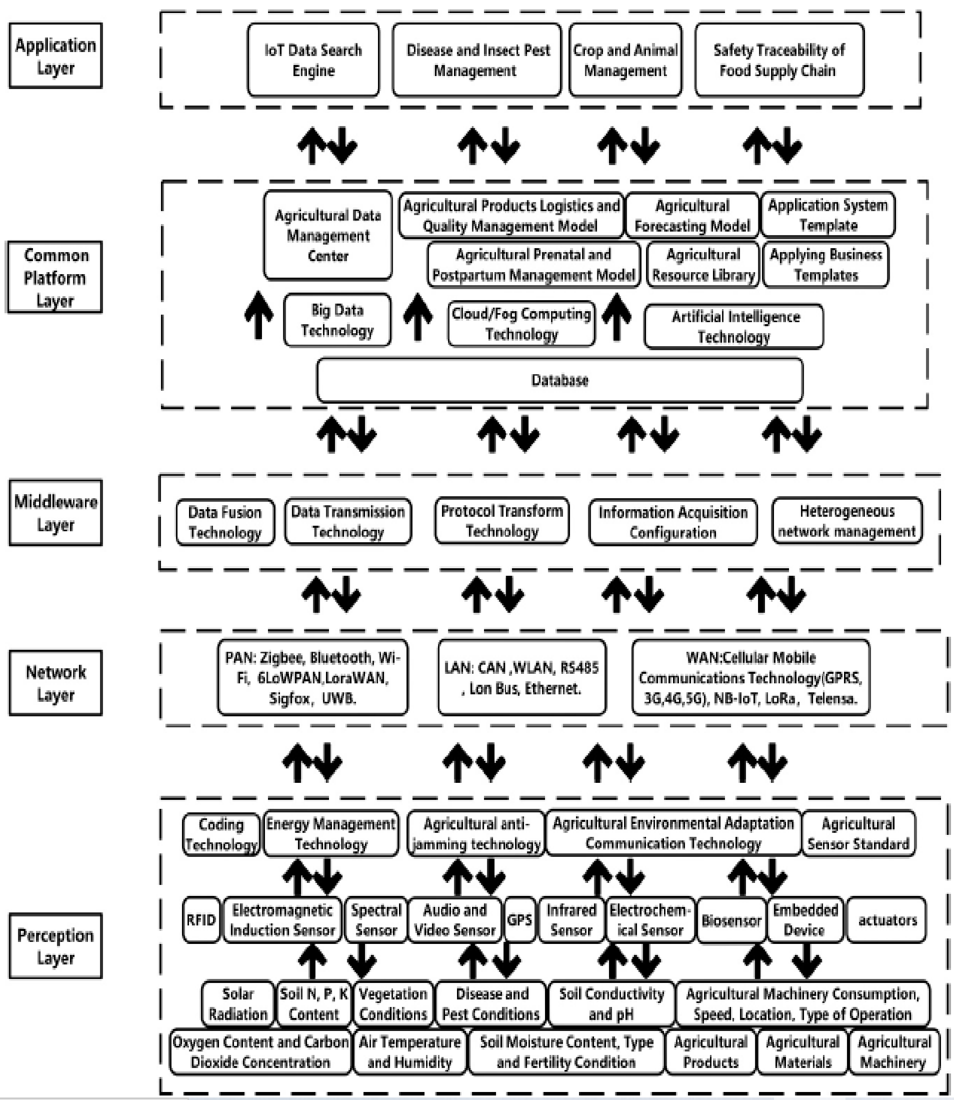

Figure 6 Structure of IoT in Protected agriculture

(1) Perception Layer: This layer comprises various sensors, terminal gadgets, agricultural machinery, wireless sensors etc. The common sensors are environmental sensors, animal and plant life information sensors. With assistance of these sensors various factors can be found out, for example, temperature, humidity, wind speed, plant diseases. The aggregate data is just prepared by the embedded device and uploaded to a higher layer through the network layer for further processing.

(2) Network Layer: The Network layer is the framework of the IoT, which incorporates a converged network formed by various communication networks and the web. It transfers information gathered from perception layer to higher layer and along with sends the control commands of the application layer to the perception layer so that sensing devices can make appropriate and suitable decisions.

(3) Middleware Layer: As in IoT devices have various different details and these can't be associated and communicated with one another which prompts issues of heterogeneity the middleware layer aggregates, filters and processes received data from IoT devices, thus reducing the time as well as cost of the overall process.

(4) Common Platform Layer: This is the layer which is responsible for decision making, storing data and establishment of various algorithms. Cloud Computing, Big Data, machine learning are the big terms that come under this part.

(5) Application Layer: After all the data gathering, analysing, creating algorithms this is the layer where implementation is done. There are also lots of intelligent platforms in this layer for the environmental monitoring and control of plants and animals like early warning, management of diseases and insect pets.

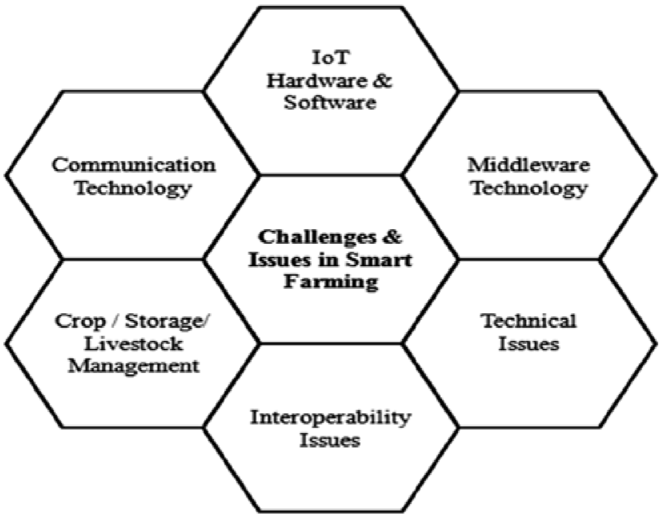

Figure 7 Common Challenges faced in Protected agriculture

They Further Discussed about the challenges of IoT technology in protected agriculture:

A. Hardware and Software Challenges: As in the perception Layer various devices will face the problems of strong solar radiations, high temperature, high humidity. Etc. The information collection nodes relay on the battery with limited power to maintain its work. Those agricultural IoT data have real time dynamic, fragmented characteristics which brings great challenges to intermediate design.

B. Network Challenge: Wired network systems cannot be used as sensors and devices that need to continuously operate under the harsh and Varied environment. Having an advantage of low cost, good networking, and flexibility wireless network become the main mode of application. But by use of normal network problems of paralysis will there. LPWAN may solve the paralysis problem of WLAN due to its long communicating distance.

C. Security Challenge: As a multi network it faces the issue of security alongside privacy protection issues, verification issues are additionally there. . As networks of different architectures need to be connected to each other, there are even greater challenges in security authentication across network architectures. . At the point when Massive information is sent to the application layer, smart handling of information, information security should be thought of. Application layer servers should have high secured authentication and encryption mechanisms.

D. Other Challenges: Thousands of devices which have huge differences in processor, memory, communication protocol are deployed in protected agriculture which results in heterogeneity issues. This problem affects the scalability of IoT in protected agriculture and hinders the use of fusion information by models.

The cost has always been a barrier to the large-scale application of IoT to ordinary farms. As it increases level of intelligence in agricultural monitoring, control, decision making thereby increases the efficiency of production, on the other hand it is changing the demand of existing agricultural workforce, causing some sort of unemployment.

2.3 Output

The Research paper talks about the advancement of technology towards IoT and its various applications especially in the field of agriculture. As with proper equipment and planning IoT can be successfully implemented in the field and provides huge benefits to farmers in terms of production, cost, time saving.

All the 5-layer discussed are interconnected to each other and proper implementation at these layers level will lead to successful implementation of the technology in the field. The Network Problem will be solved by using LAN instead of WLAN. algorithms and other analytics are to be done by Machine Learning and Artificial Learning.

CHAPTER 3

COMMUNICATION

3.1. Cloud Communication using Thingspeak:

Cloud communication is a way of communicating with the help of Internet rather than over a wired/ standard telephone network. The cloud base provides a platform to store any collected data in a real time basis, and be able to securely access the collected data from any part of the world. Using this mode of communication absolutely removes the need of installation of physical installation of wires, thus increasing the accessibility and reducing the time and cost aspect of both, setup and maintenance over time. The data storage is hosted by a third-party outside of the organization using them, and they are accessed over the public Internet.

The requirement of cloud communication in this project arises because of:

1. The need to collect various parameters of the soil and constantly upload the data to a place where it can be constantly monitored without a farmer physically being on the field.
2. Applying ML algorithm to constantly test the data being collected and update the model.
3. Ensuring wireless transfer of data from remote areas.

3.1.1 ThingSpeak – What and Why?

- ‘ThingSpeak’ is an IoT analytics platform based on cloud that allows you to aggregate**,** visualize and analyse live data streams in the cloud.
- It provides instant visualizations of data posted by your devices to ThingSpeak.
- It can be accessed from anywhere in the world, protected by a unique API key and a channel number.
- It has the ability to execute MATLAB code in ThingSpeak, can perform online analysis and processing of the data as it comes in.
- It can configure at what intervals do we need our data, as well as use MATLAB alongside to implement ML algorithms in real time.
- It has some inbuilt applications that can react to any abnormal data, and inform the host through a trigger link.

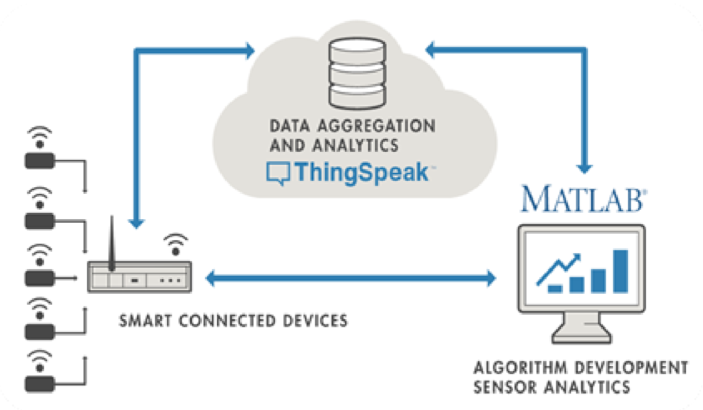

Figure 8 Working of ThingSpeak

Figure 9 Flowchart of ThingSpeak configuration

3.1.2 Field Variables:

- It represents the various variables that we intend to measure through a channel.
- Every field variable can be graphically viewed in real time.
- The names of the field variables are case sensitive, in the case that the code in the microcontroller should have the exact same characters of the variable names as it's defined on the ThingSpeak platform.
- In this project the field variable consists of the various parameters to be measured.

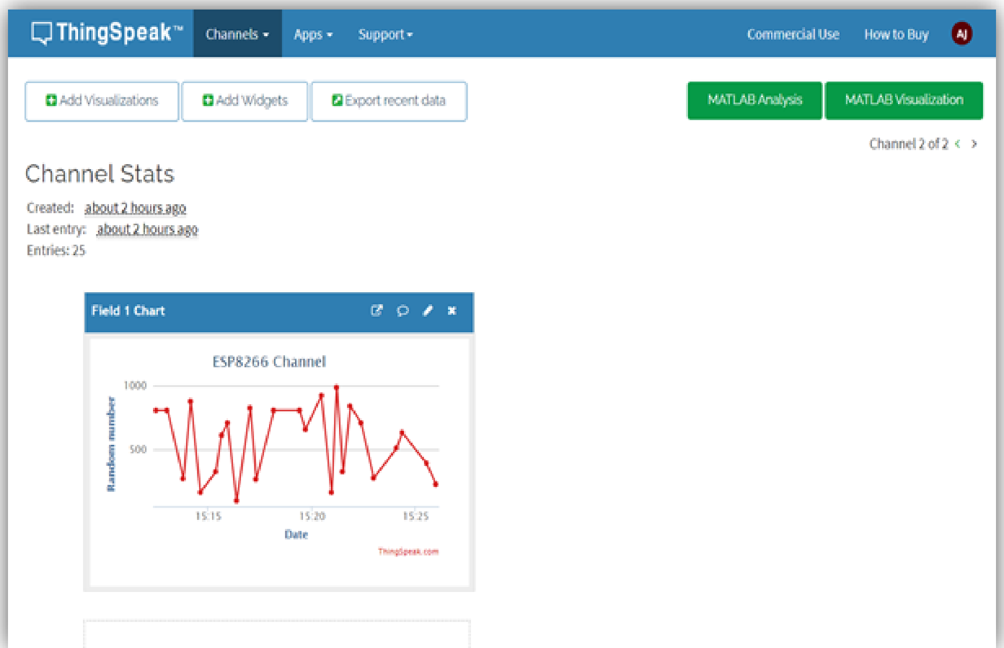

Figure 10 Random number as a field variable on ThingSpeak

3.1.3 ThingHTTP

ThingHTTP enables communication among devices, websites, and web services without having to implement the protocol on the device level. You specify actions in ThingHTTP, which you trigger using other ThingSpeak apps such as TweetControl, Time-Control, and React.

- Using ThingHTTP requires a prior trigger link to be created.
- In the course of this project, the trigger link is created with the help of 'Webhooks' platform (discussed in a later section in detail).
- The triggering URL is then fed into the ThingHTTP server, which helps connect that URL to our channel.
- This is again securely protected with the help of the secret API key.
- Solely using ThingHTTP does no work, it has to be combined using some other triggering app that directs ThingHTTP link to be opened at a defined condition by the admin.

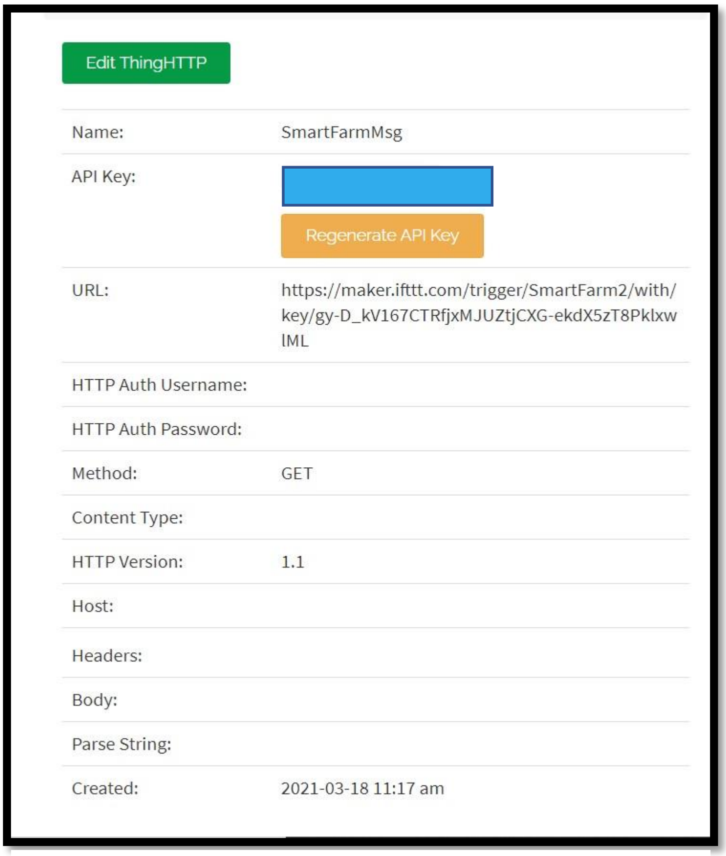

Figure 11 ThingHTTP Window

3.1.4 React (ThingSpeak)

- React works with ThingHTTP, ThingTweet, and MATLAB Analysis apps to perform actions when channel data meets a certain condition.
- In this project, React is being used to trigger the link that has been mentioned in the ThingHTTP app.
- The condition on which react operates can be anything based on the field variables, for example, if the soil moisture percentage exceeds 60%, trigger the link, which further triggers another action (ex. A text message on a mobile phone)
- If you use the React app with the Test Frequency set to On Data Insertion when using Bulk-Write JSON Data, your React is triggered once per bulk-update request. It is triggered when Condition matches Condition Type, even if the condition is matched more than once.
- This ensures non-repetitiveness of trigger.

Name:	React 2
Condition Type:	Numeric
Test Frequency:	On data insertion
Last Ran:	2021-03-18 14:37
Channel:	SmartFarm
Condition:	Field 3 (Soil Moisture) is greater than 60
ThingHTTP:	SmartFarmMsg
Run:	Only the first time the condition is met
Created:	2021-03-18 11:18 am

Figure 12 React Window

3.1.5 API Authentication

- An API key is a simple encrypted string that identifies an application without any principal.
- They are useful for accessing public data anonymously, and are used to associate API requests with your project to access.
- The data lying on a cloud platform can be kept private with the help of these API Keys.
- <u>Write API Key</u> – A 16 digit code that allows an application to write data to a channel
- <u>Read API Key</u> – A 16 digit code that allows an application to read the data stored in a channel
- The same write API key has to be mentioned while programming the microcontroller which gives it the access to write the collected data from sensors, onto the ThingSpeak platform.

3.2 App development using MIT App Inventor

- MIT App Inventor is a web application integrated development environment that was originally provided by Google and is now maintained by the Massachusetts Institute of Technology (MIT).
- MIT App Inventor is an intuitive, visual programming environment that allows everyone to create fully functional apps for smartphones and tablets.
- It allows newcomers to computer programming to create application software (apps) for two operating systems (OS): Android, and iOS.
- It uses blockchain-based programming languages based on Google Blockchain (Fraser, 2013) and Star logo, TNG (Bezel and Clofer, 2007) and Scratch (Resnick et al., 200; Maloney, Resnick, Rusk, Silverman, and Eastmond, 2010), empowers anyone to create a mobile phone app to meet their needs.

3.2.1 Why MIT App Inventor?

- First, it does not require any kind of advanced coding and easy to use.
- It lets you develop apps for Android phones using a web browser and either a connected phone or an emulator.
- Its servers store your work and help you track your projects.
- This website provides you with all the support you need to learn how to create your own apps.

3.2.2 Interface

The MIT App Inventor user interface consists of two main editors: Design Editor and Block Editor. There is a drag and drop interface to hold elements of the user interface (UI) of a design editor or designer application (Fig 13). Blocks Editor is an environment in which app explorers logically visualize their apps using coded blocks of colours coded together like puzzle pieces to describe a program (Fig 14). To help with development and testing, App Inventor provides a mobile app called App Inventor Companion (or simply “Companion”) that developers can use to test and adjust the real-time behaviour of their apps. That way, anyone can quickly create a mobile app and quickly repeat and test it.

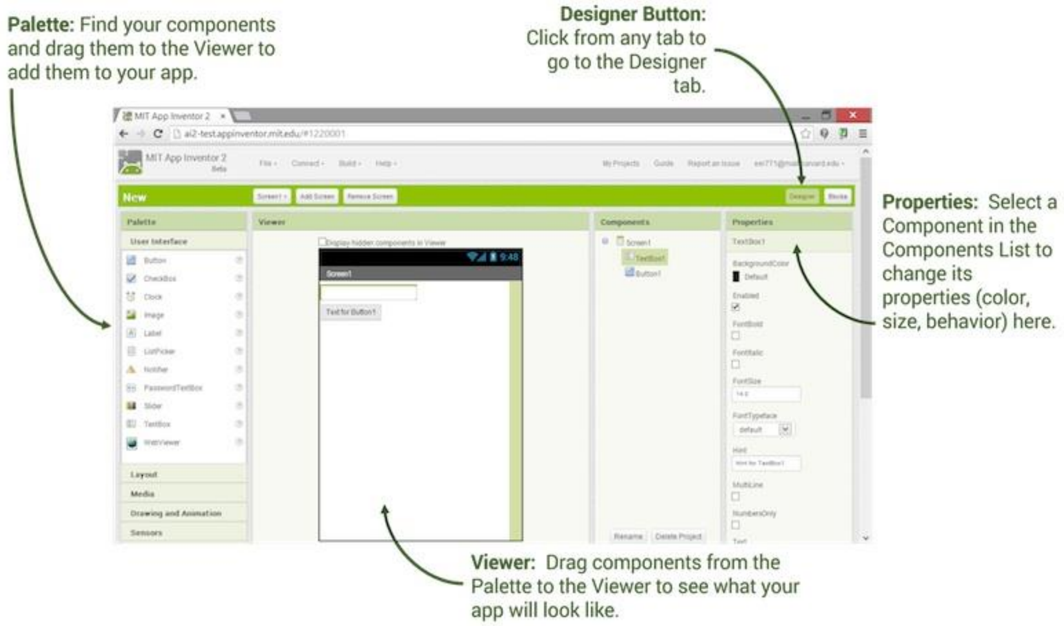

Figure 13 App inventor's Designer Tab

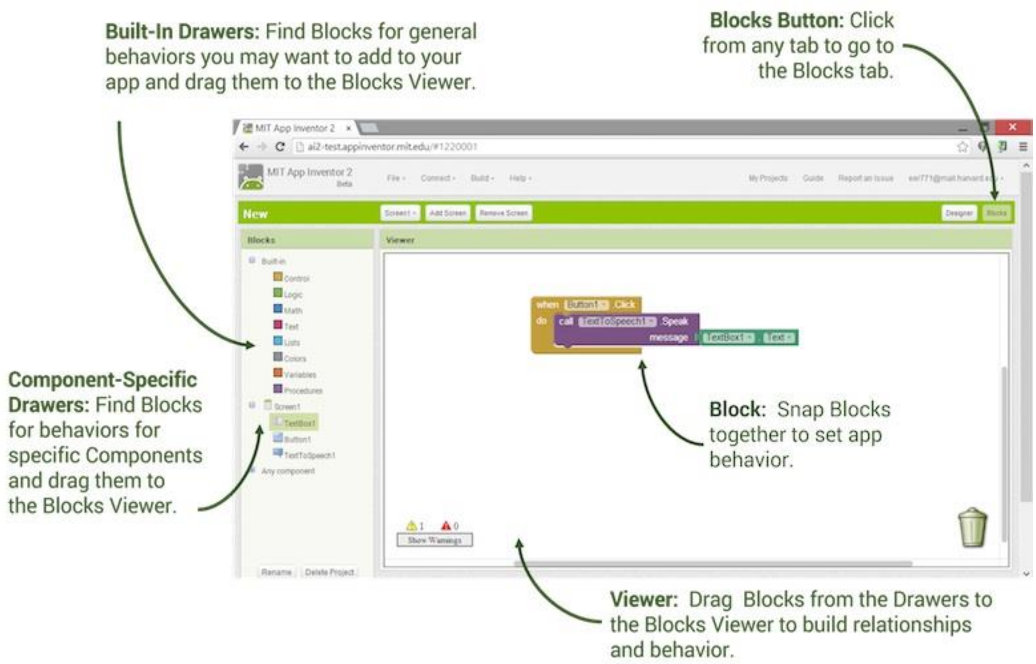

Figure 14 App Inventor's Block Editor

3.2.3 Design Goals

The main objectives of the MIT app inventors are to introduce mobile app minutiae in an educational context. Prior to its release, most minutiae environments for mobile applications were difficult, wieldy only through system-level or embedded programming or both. Even with Google's Android operating system and Java programming language, designing a user interface was a ramified task. In addition, the use of the platform requires Java syntax and economics and the worthiness to debug Java compiled errors (e.g. incorrect spelling variables or semicolons in the wrong place). These challenges created barriers to app researcher target demographics, computer science access. We transiently discuss and discuss the diamond objectives of the App Finder project, in particular the use of components to overcome the complexity of platform policies and the complexity of the underlying programming language. These objectives can be remoter described with the aim of enabling search through the structuring of visual language and fast, repetitive diamond in the mental model of young developers.

The components in the MIT App Inventor are vital abstractions. Components reduce interoperability with platform-specific programming using programming interfaces (APIs) and reduce state management details of device hardware. This allows the user to think well-nigh the problem rather than the minutes the developer needs. Components are made up of three main components: properties, methods, and programs. The properties are readable and / or readable by the app developer and the status of the tenant elements.

3.2.4 Block Coding:

In MIT App Inventor, users law making using policies using block-based programming languages. App Explorer has two types of blocks: seated woodcut and component block. The vital atoms and operations in the seated woodcut library are usually traditional in other programming languages such as Boolean, wire, number, lists, mathematical operator, comparison operator, and tenancy spritz operator. Developers use component blocks (properties, methods, and events) to respond to system and user events, to interact with device hardware, and to retreat the visual and behavioural aspects of components. All program logic is built on three top-level woodcut types: Global Variable Definition, Process

Definition, and Component Event Handler. Global variables provide designated slots for storing program statuses. Procedures define common behaviours that can be called from multiple locations in the code. When an event arrives on the device, it triggers the corresponding application behaviour suggested in the event block. The event handler block can refer to a global variable or process. By limiting top-level block types, there are fewer factors to argue about.

As we needed to send the sensor data from ThingSpeak platform to MIT App inventor, we have used the component called Web viewer.

3.2.5 Web Viewer

'Web Viewer' allows the loading of web pages or websites in the app. To do that, we can simply drag the web viewer element from palette to the Screen 1 window and set the web viewer's home URL property to the webpage we want to see when the app boots. For the project, we need data from temperature, humidity, and soil moisture sensors for every text box, a corresponding ThingSpeak server link has been attached and the same applies for the respective graphs.

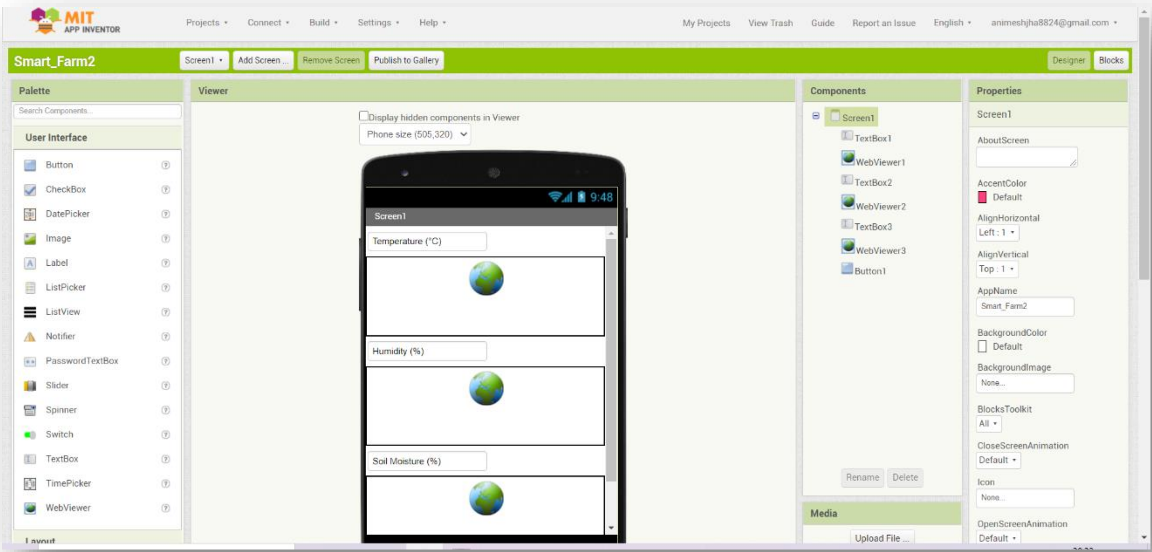

Figure 15 Interface of Mobile App in MIT App Inventor (A)

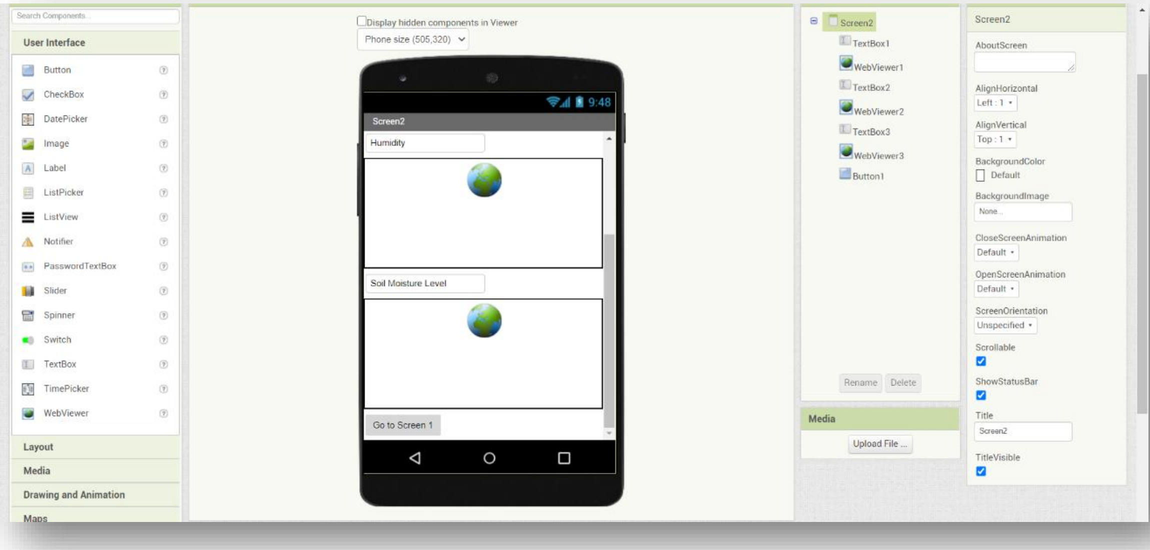

Figure 16 Interface of Mobile App in MIT App Inventor (A)

3.2.6 MIT AI2 Companion

The MIT AI2 Companion app enables real-time debugging on devices unfluctuating via Wi-Fi or Universal Serial Bus (USB). In addition, the user can use the "On Computer" emulator for Windows, MacOS and Linux. You can use App Finder without downloading anything to your computer.

User has to just install the MIT App Finder Companion app on your phone, tablet or supported Chromebook to test live on your smartphone or tablet device. Once Companion is installed, you can unshut a project in the app finder on the web, unshut a companion on your device, and test your apps when you're ready.

3.3 IFTTT (If This Then That)

3.3.1 Introduction

‘If this then that’ (IFTTT) is a service that allows the user to program a response to a variety of world events. IFTTT has a long list of instances where it can respond, all searchable via the Internet. Rain is forecast for tomorrow in the weather underground report. The second is that someone tagged the user in a photo on Facebook. There is a long list of similar reactions that are executable all over the internet. An example reaction would be to send an email to the user stating that there is a forecast of rain or to copy the photo mentioned above in the user's archive. IFTTT has partnerships with hundreds of service providers that provide event notifications to IFTTT and execute commands executing responses, but some programs and command interfaces are only public APIs.

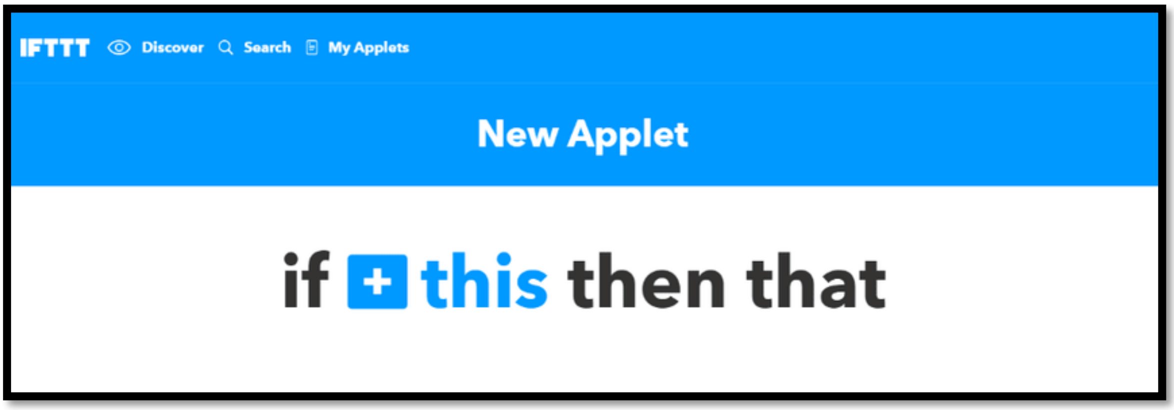

Figure 17 Interface of IFTTT

IFTTT helps you connect all your apps and devices. When you sign up for a free account, you may be able to work together to do things that some apps and devices can't. Applets and connections can be built by services or users within the IFTTT Platform.

Here's how it works:

1.Create an account.

2.Browse the IFTTT website or app to find something that interests you.

3.Connect the services that are involved in the Applet or connection. (Applet: - An applet that adds together two or increasingly apps or devices. This enables us to do something which they cannot do on their own apps or devices.)
4.Find increasingly Applets and connections, and repeat!

Some of the services of IFTTT used for our project includes Webbooks, Android SMS, Gmail.

3.3.2 Webhooks

A webhook in web minutiae is a method of enhancing or interchange policies with a custom call back of a webpage or web application. These call-backs may be maintained, modified, and managed by third-party users and developers who do not need to be associated with the original website or application. The word "webhook" was coined by Jeff Lindsay in 200 programming from the computer programming term hook. Webbooks are streamlined messages sent from apps when an event occurs. They have a message - or payload - and it is sent to a unique URL - basically the app's phone number or address. This message, we called as triggering.

3.3.3 Android SMS

Android SMS is a vital service that allows you to receive short message service (SMS) messages on your device and send messages to other phone numbers.
And other services like Gmail

3.3.4 Steps involved to enable the SMS/Gmail alert

Step1: Login to the IFTTT on www.ifttt.com

Step 2: Create new applet. For that click on My applet then following page can be shown.

Figure 18 Step 2 of IFTTT

Step 3: Click on this and select the Webhooks service. And give the triggering event name and other variables. And then click on Create Trigger.

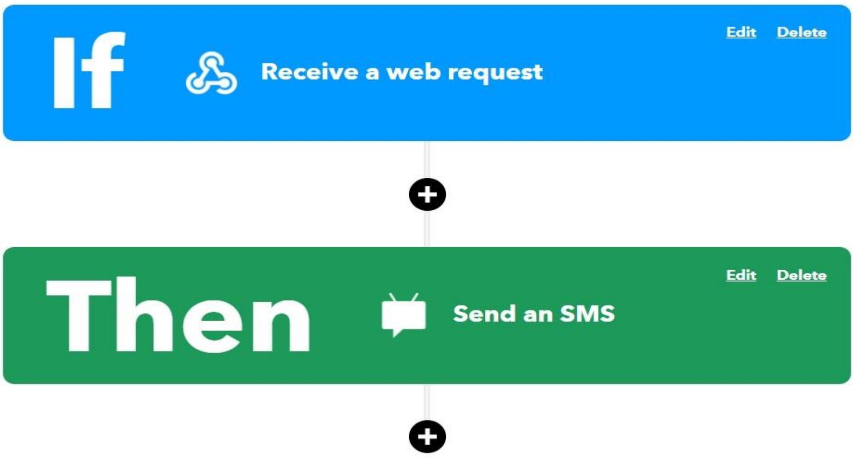

Figure 19 If ‘Webhooks’ then ‘SMS’

Step 4: Now click on Android SMS / Gmail and write message with variables and Phone number / Gmail address.

Step 5: Copy the Webhooks link. And then open ThingSpeak and in app section click on ThingHTTP. Create new and paste the webhook link their and click create.

Step 6: After setting ThingHTTP, again click on app and select React. And select the necessary condition. Every time the condition satisfies the SMS will be triggered.

CHAPTER 4

METHODOLOGY

4.1 CAD Model

4.1.1 Initial CAD Model

The entire system of components is incorporated in a 3d printed cube and the cube is designed using Auto-CAD software.

The specifications of the cube are mentioned as below

- The dimensions of the cube 310*260*100 mm^3 is required for the system to be assembled properly.
- A hole is provided at the base of cube to accommodate wires running down to the ground.
- A hollow pole is attached to provide protection to the wires and to provide sufficient height for the IoT Cube.
- The length of pole is 1000mm and internal diameter is 50mm.
- The solar panels are mounted on top of the cube while other components of the power supply system are assembled inside of the cube as shown in fig.
- The angle of tilt of solar panel is same as the latitude of the location i.e., 20 degrees.
- Wires run down through the pole into the ground for the sensors which are to be inserted in the soil.
- Sensor heads which aren't required to be inserted in the soil are fixed on the wall of the cube to get exposure of the environment as shown in figure.
- The screenshot of model is attached alongside the different views of the system for better understanding.
- The door is opening from the front as shown which allows to make any changes, improvements, tests or any repair works on the system.

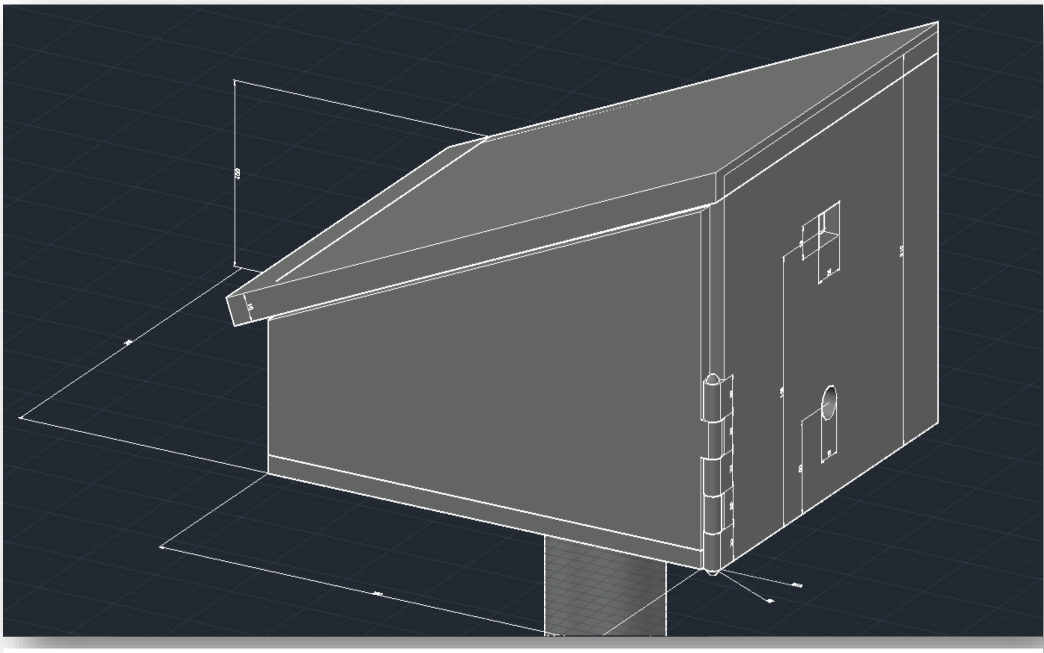

Figure 20 Front door initial model

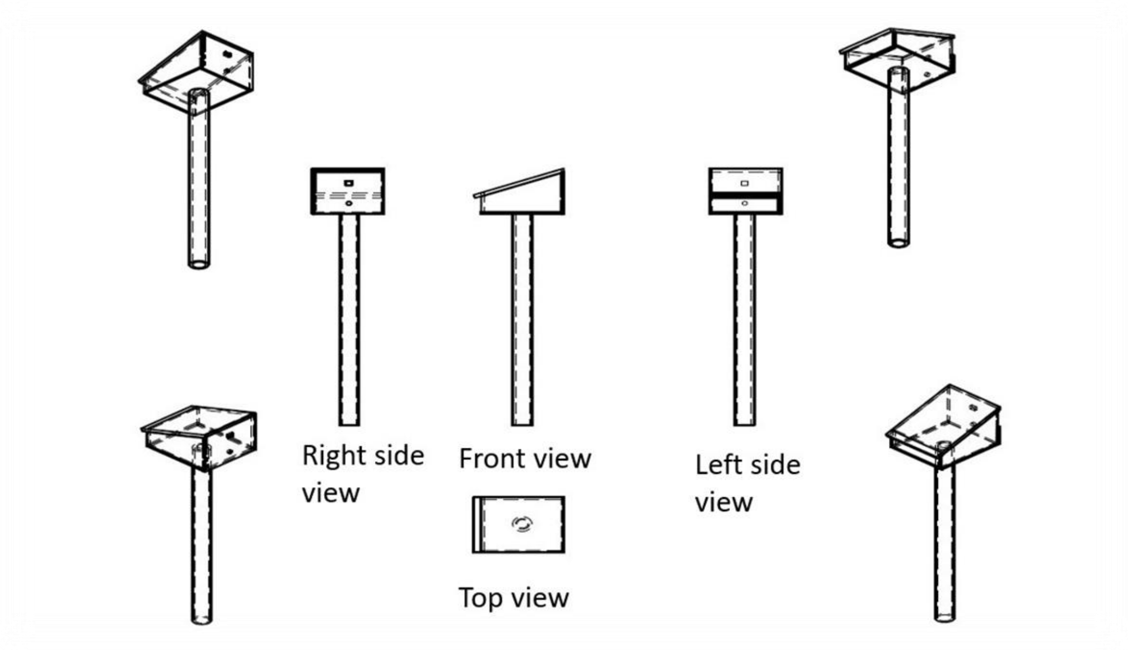

Figure 21 Different views of front door model

4.1.2 Modified CAD Model

The initial model has been improved further by changing the alignment of the door form front opening to the top opening which then provides easy access to the internal parts and importantly all at once whereas in the previous model, there is no easy access to all components at once as they lie on the floor bed which demanded a top view.

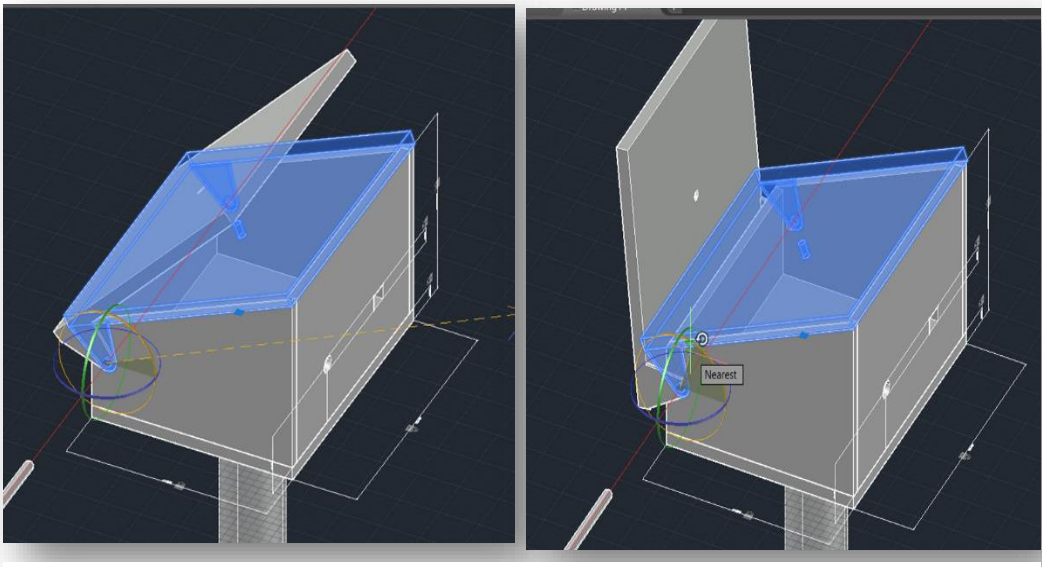

Figure 22 Modified Top door model

- And also, a root type of system has been incorporated at the base of the pole so that it shall protect sensors from unwanted external forces like object impacts or human/animal forces.
- This protects the sensor ends while, through grooves, it still allows the water to seep and stay in same condition as its neighbouring soil parts.
- This also increases the stability of the system during winds or heavy water flow in the farm.
- These improvements made the system much more reliable and stronger.

- The root system at the end has a total of six tentacle like structures which dig into the soil to provide the required grip as shown in the figure below.

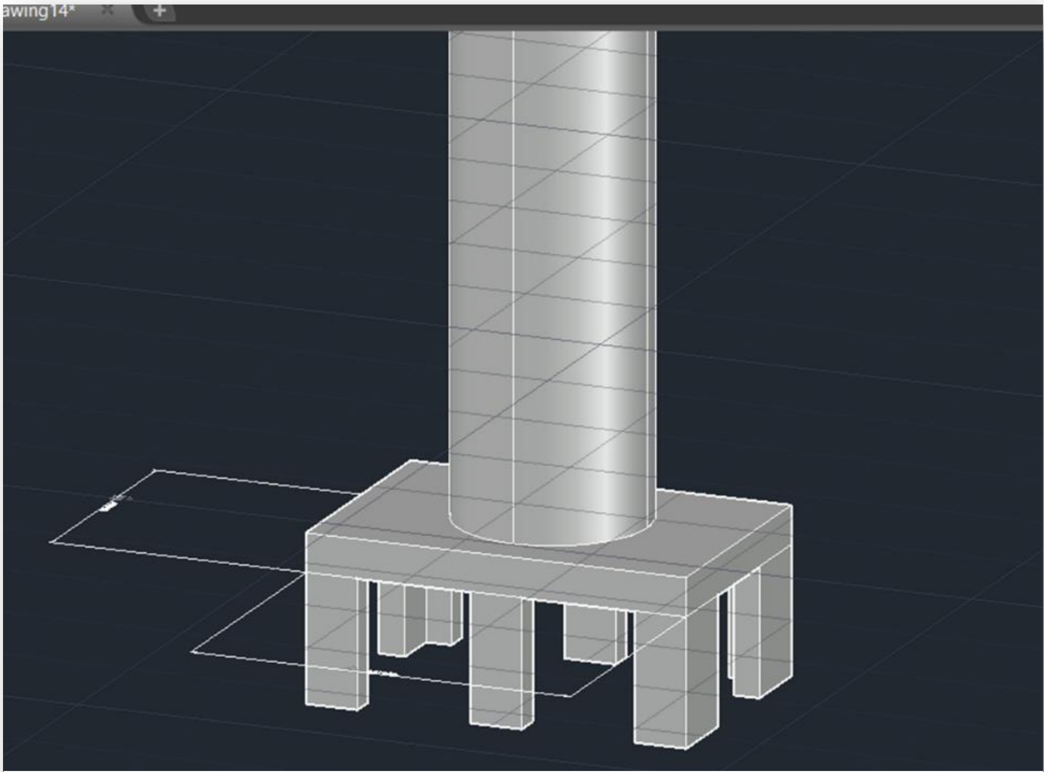

Figure 23 Root system of Modified Model

- The power supply to the cube is being provided using solar panel which was explained in the next section.
- The solar panel is mounted on the top of tilt face of the cube and the connections are drawn from that wire to inside of the cube.
- There is a small hole on the top face door of the cube which allows the connecting jump wires to get inserted inside of the cube from the solar panel.
- Then all the remaining components of the power supply system are then connected inside of the cube so that they get protected from the outside environment ensuring their safety in the long run.

- So, this is how the overview of final cad model would look like.

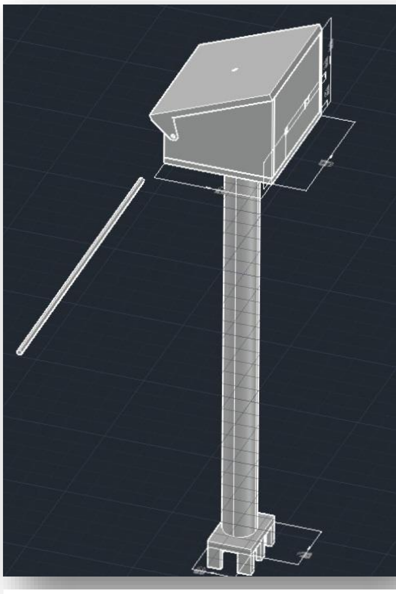

Figure 24 Overview of final model

- And the various views of the finalised IoT Cube CAD model are shown in the figure below.

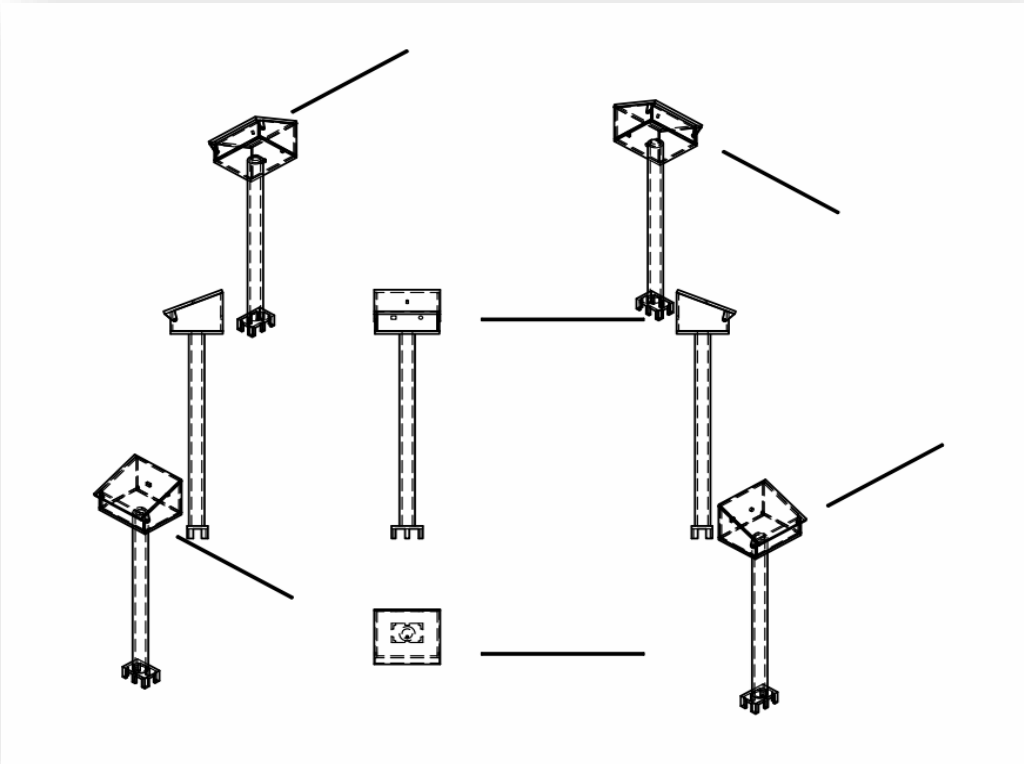

Figure 25 Different views of final model

4.2 Power Supply

A summarised calculation for the power supply for a single Cube is as follows:

- The power requirement of whole system is being provided by solar energy.
- The voltage required for the working is 9V while the current requirement is approximately 1 ampere in total. This gives the load of 9W.
- Considering the average duration of sun light be for 8 hrs a day then battery should provide power for 16hrs in a day.
- Taking the time of usage into consideration we require 9W*2hr=18Whr per day.
- So, the considered battery has the capacity 1500mAh which has voltage rating 7.4V so it can power the whole system. Thus 2 batteries are required to give 18/7.4=2432 approximately 3000 mAh per day considering losses.
- Solar panel should power both load and the battery as well. This can be provided using one 12V, 6amp solar panel which is sufficient power for our system.
- A charge controller is used in between to regulate the direction of flow of current and stabilise the fluctuations.
- Thus, the components required are one 12V,10Watt solar panels, TIDA-010042 MPPT charge controller, two 1500mAh, 7.4V, 3.25Watt Li-ion rechargeable batteries along with all the sensors.
- The load should be connected in parallel with the battery.

All components are decided considering losses and availability of standard rated equipment.

4.3 Component List

Sr. No.	Components
1	Arduino UNO and Nodemcu
2	RS485 Soil Moisture Sensor
3	DHT22 Temperature and Humidity Sensor
4	ESP8266 WiFi Module
5	2*1500mAh 7.4V Li-Ion Battery
6	Breadboard
7	12V, 10W Solar Panel
8	GSM Module (for SMS Alerts)
9	LORA Module
10	Jumper Wires
11	Resistances

Table 2 Component List

4.4 Circuit Diagram

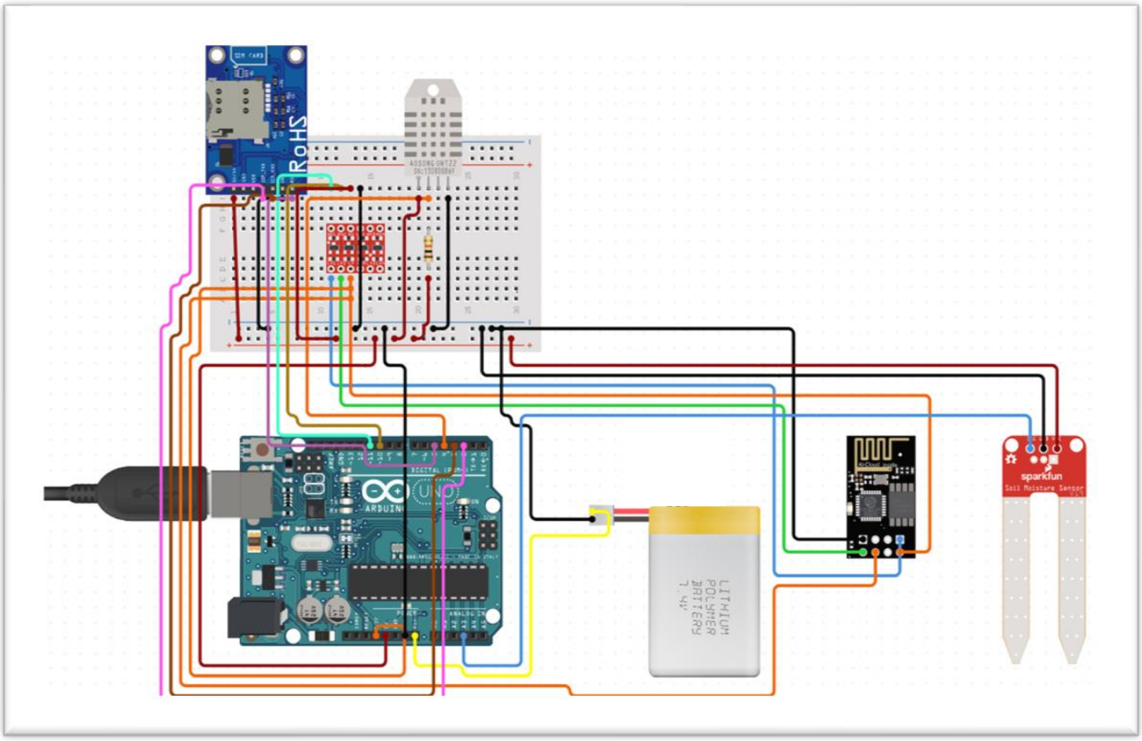

Figure 26 Circuit Diagram

4.5 Master Code (in Arduino IDE)

```
#include <DHT.h>  // Including library for dht
#include <ESP8266WiFi.h>

String apiKey = "****************";     //  Enter your Write API key from ThingSpeak

const char *ssid =  "******";     // replace with your wifi ssid and wpa2 key
const char *pass =  "*******";
const char* server = "api.thingspeak.com";

#define DHTPIN D2          //pin where the dht11 is connected

DHT dht(DHTPIN, DHT11);
WiFiClient client;

void setup()
{
       pinMode(A0, INPUT);
       Serial.begin(115200);
       delay(10);
       dht.begin();
       Serial.println("Connecting to ");
       Serial.println(ssid);

      WiFi.begin(ssid, pass);
      while (WiFi.status() != WL_CONNECTED)
     {
            delay(500);
```

```
        Serial.print("."); //printing "…" until WiFi is not connected
    }
    Serial.println("");
    Serial.println("WiFi connected");

}

void loop()
{

    float h = dht.readHumidity(); //reading humidity value
    float t = dht.readTemperature(); //reading temperature value
    float msvalue = analogRead(A0); //reading soil moisture value in range 0-1024
    float msvalue1= ((1024-msvalue)/874)*100; //normalising based on calibration

         if (isnan(h) || isnan(t))
           {
              Serial.println("Failed to read from DHT sensor!");
              return;
           }
         if (isnan(msvalue) )
           {
              Serial.println("Failed to read from Soil Moisture Sensor!");
              return;
           }
                if (client.connect(server,80))   //   "184.106.153.149"
              {
                   String postStr = apiKey;
                   postStr +="&field1=";
                   postStr += String(t);
```

```
                        postStr +="&field2=";
                        postStr += String(h);
                        postStr +="&field3=";
                        postStr += String(msvalue1);
                        postStr += "\r\n\r\n\r\n";

                        client.print("POST /update HTTP/1.1\n");
                        client.print("Host: api.thingspeak.com\n");
                        client.print("Connection: close\n");
                        client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
                        client.print("Content-Type: application/x-www-form-urlencoded\n");
                        client.print("Content-Length: ");
                        client.print(postStr.length());
                        client.print("\n\n\n");
                        client.print(postStr);

                        Serial.print("Temperature: ");
                        Serial.print(t);
                        Serial.print(" degrees Celcius, Humidity: ");
                        Serial.print(h);
                        Serial.print("Soil Moisture: ");
                        Serial.print(msvalue1);
                        Serial.println("%. Send to Thingspeak.");
                    }
        client.stop();
        Serial.println("Waiting...");
  // thingspeak needs minimum 15 sec delay between updates
  delay(1000);
}
```

4.6 Model Run

The above shown circuit was constructed physically and was tested on a home soil sample, the complete working starting from collecting data from the sensors to sending that data on the cloud platform to providing alerts in case any limit is surpassed has been demonstrated further with the assistance of clicked pictures (Based on Model Dry Run performed on 18-03-2021 at 8:06 pm).

A. The circuit can be seen in the figure below, where the soil moisture sensor is currently moisture free (At the time instance 8:06 pm)

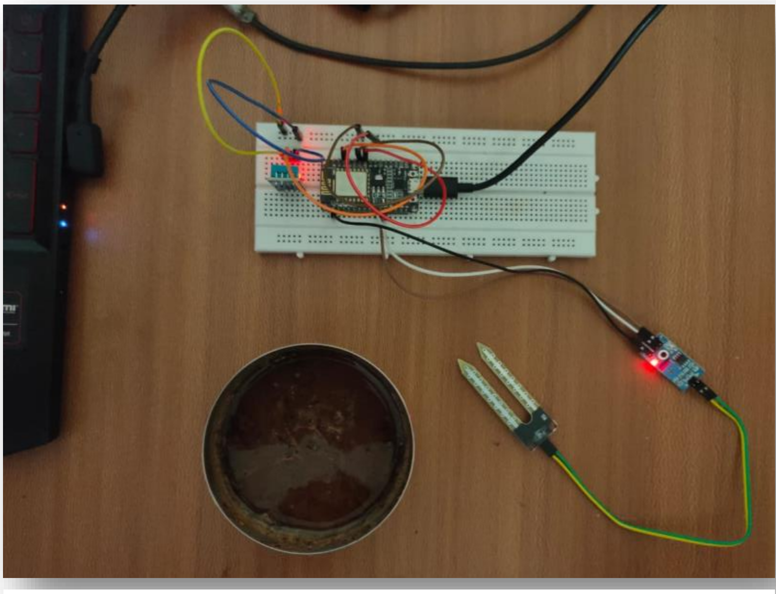

Figure 27 Circuit Testing - A

The next figure demonstrates the mobile application that has been built to monitor data in real-time. The 1st screen of the app digitally shows the readings of various sensors. A point to be noted is that these readings can be seen from anywhere around the globe with just the basic requirement of a basic 2G internet connection. The application can be installed on any Android operated mobile device, using the .apk setup file that has been created in the project.

At the instance of time 8:06 pm, the value of soil moisture percentage can be seen as 1.5%, which is quite a justified value near to 0%, the error being because of the pre-exposure of sensor to the soil moisture in near past.

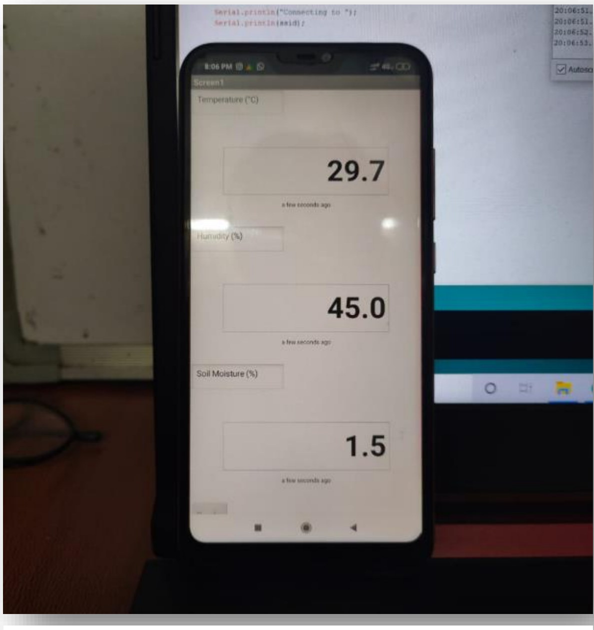

Figure 28 Mobile Application Testing - A

B. The next figure shows the instance where the soil moisture sensor is dipped into a high moisture content soil sample (At the time instance 8:07 pm), and ideally this should produce an alert stating that the moisture level has suddenly risen to a high value, beyond limits.

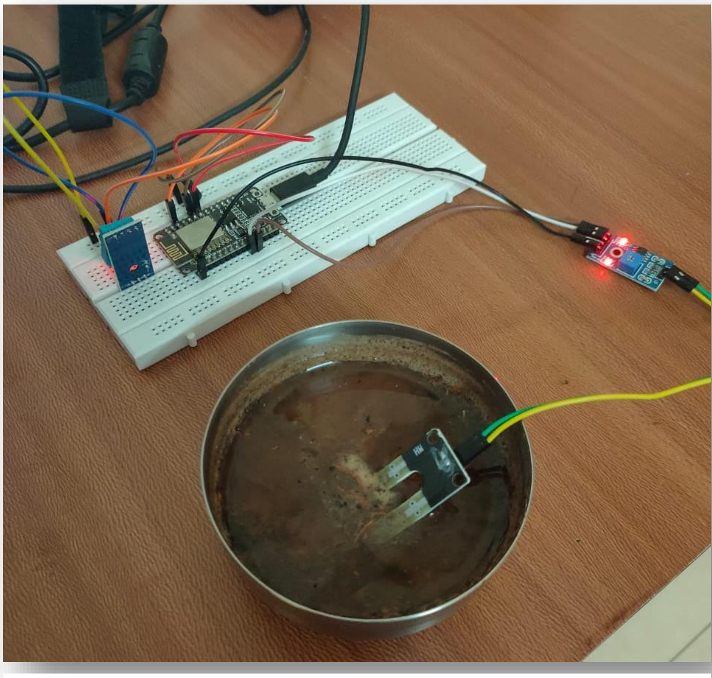
Figure 29 Circuit Testing - B

The screen figures shown further depicts the mobile application displaying the increased soil moisture level in two form of representations, digitally on the first screen and on the second screen, the graphical form of data is shown in a Value vs Time Graph. (The mobile shows the time instance of 8:07 pm, indicating quick responsiveness in real time)

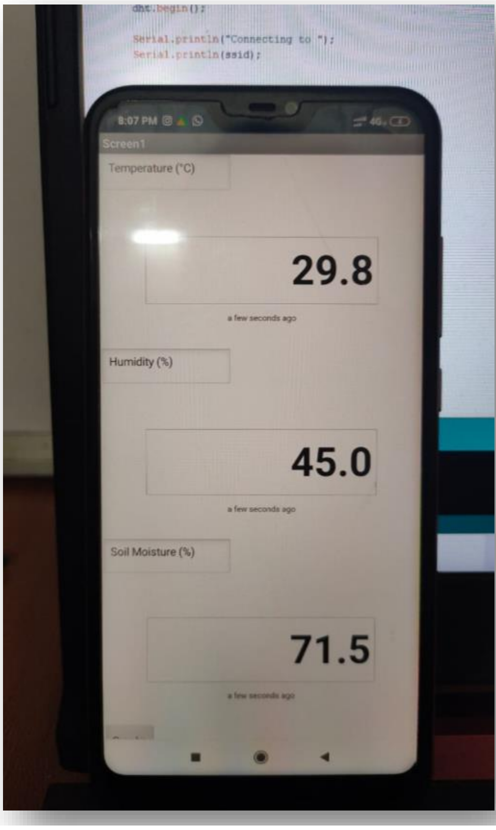

Figure 30 Mobile App Testing - A (Digital)

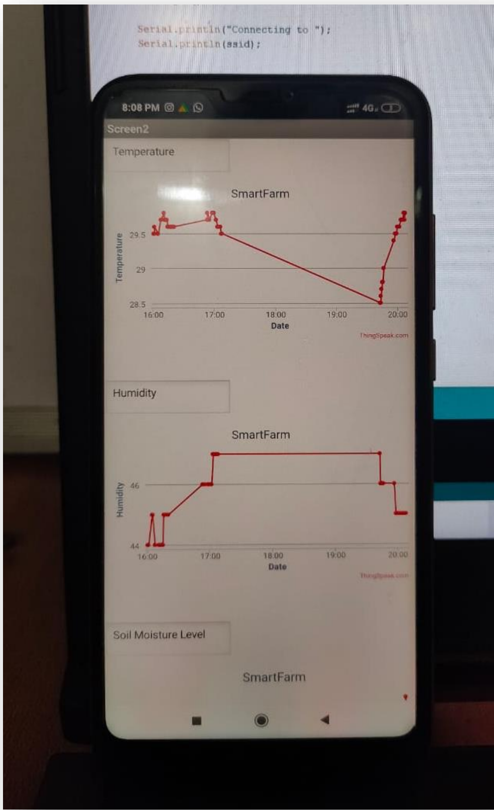

Figure 31 Mobile App Testing - B (Graphical)

C. These figures show the two ways in which alerts can be sent on the farmer's mobile phone, either to their email ID or to their mobile numbers as text SMS, (An additional call feature can also be added as an increased scope, but that comes with an increase in the capital cost). The time instance of recieveing the alert is 8:08 pm, again indicating quick responsiveness to the situation.

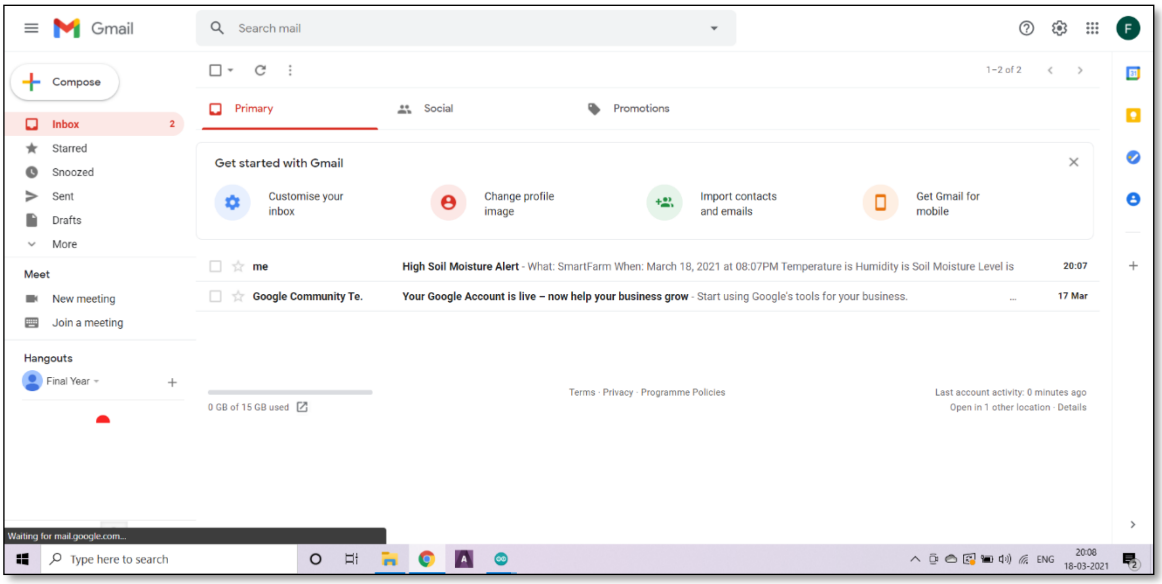

Figure 32 Gmail Notification Alert

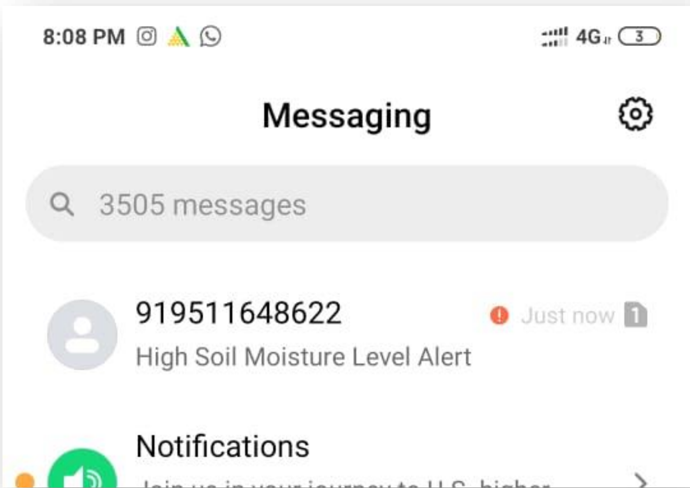

Figure 33 Android SMS Alert

4.7 Machine Learning Algorithm:

4.7.1 Objective

With help of Machine Learning Algorithm, we are going to set the threshold values for the levels (low, medium and high) for data values which we are getting from sensors and can also predict the future trends.

Unsupervised Learning: It is a type of machine learning algorithm in which labels or output are unknown while it focuses on finding the patterns and form clusters and gaining insight from the dataset.

Clustering: Clustering or cluster analysis is an unsupervised learning problem. This method used as data analysis for discovering interesting problems like grouping of customers based on their choices and behaviour. Basically, this find homogenous subgroups within data such that data points in each cluster are as similar as possible according to the similarity measure set such as distance or correlation-based distance.

4.7.2 K-Mean Clustering Algorithm:

K mean algorithm is an iterative algorithm which tries to partition the dataset into k pre-defined distinct non overlapping subgroups(clusters) and each data belongs to only one group. Its tris to make intra cluster data points as similar as possible.

Assign data to cluster: It assigns data points to a cluster such that the sum of squared distance between the data points and the cluster centroid is at minimum. The less variation we have within clusters more homogenous will be data in the clusters.

Working/Steps of K Mean Clustering algorithm: -

1. Specify the numbers of cluster k
2. Normalize the data sets as data set have very ranging values.
3. Initialize the centroids by first randomly selecting k data points as centroid

4.Find the distance of data points with the centroids and then assign points to clusters on basis of their distance.

5.Compute the centroids for the clusters by taking average of all the data points that belong to each cluster.

6. Again iterate the step 3 and step 4 until there is no change to the centroids. (assignment of data points to clusters isn't changing.)

Formulas and Mathematical Expressions:

1) Objective Function of K-Mean Clustering is defined as:

$$J = \sum_{i=1}^{m} \sum_{k=1}^{K} w_{ik} \|x^i - \mu_k\|^2 \tag{1}$$

Here $w_{ik} = 1$ if data point x belong to the k_i cluster otherwise it will be considered to be zero.

2) As we have to calculate minimum distance so it is done in two steps i.e E-step and M-Step.

E step is basically for assigning data points to the cluster.

M Step is for computing the centroid of each cluster.

i) Firstly, minimize the objective function(J) by differentiating it with respect to w_{ik} by treating u_k as constant. Then update the cluster assignments. This Step is called E-step.

Mathematically, E Step is:

$$\frac{\partial J}{\partial w_{ik}} = \sum_{i=1}^{m} \sum_{k=1}^{K} \|x^i - \mu_k\|^2$$

$$\Rightarrow w_{ik} = \begin{cases} 1 & \text{if } k = argmin_j \|x^i - \mu_j\|^2 \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

ii) Now differentiate the objective function(J) with respect to u_k by taking w_{ik} as constant. Then redo the cluster assignments step from previous Step. This step is called M -Step.

Mathematically. M-Step is:3

$$\frac{\partial J}{\partial \mu_k} = 2\sum_{i=1}^{m} w_{ik}(x^i - \mu_k) = 0$$

$$\Rightarrow \mu_k = \frac{\sum_{i=1}^{m} w_{ik}x^i}{\sum_{i=1}^{m} w_{ik}} \tag{3}$$

4.7.3 Algorithm Demonstration:

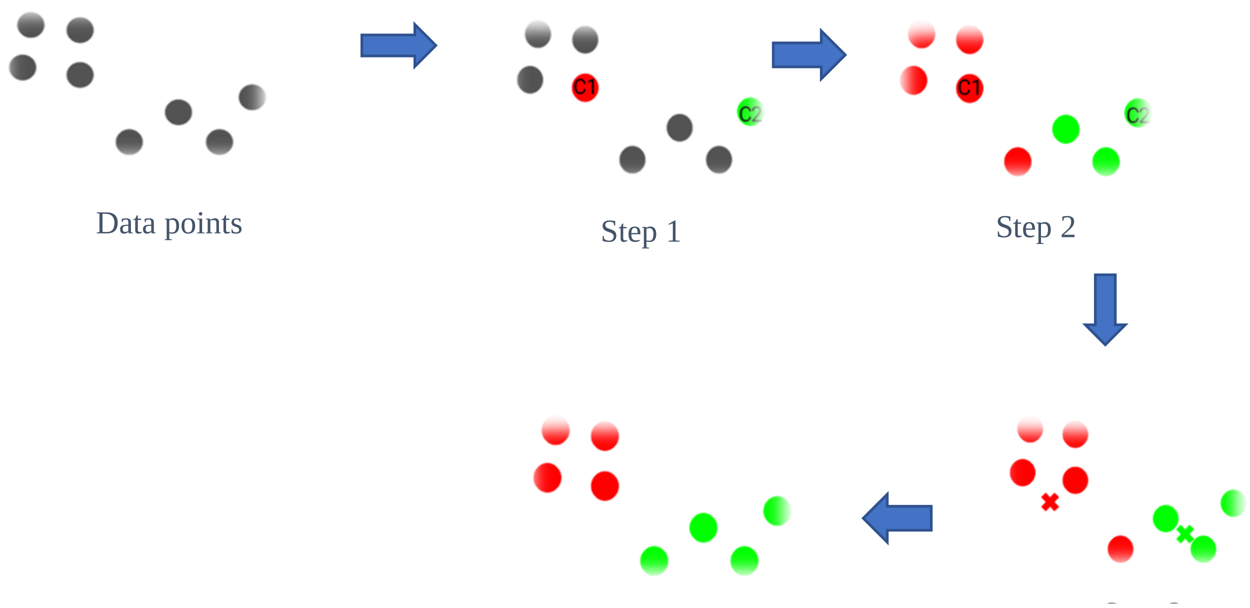

Figure 34 Algorithm Demonstration

Here in above example, we have 8 data points and we want to divide the data points in two clusters. So, Procedure for division of data in clusters as follows: -

Step 1: Randomly picking 2 data points as centroids. Here in this case k =2

Step 2: Now Calculating the distance of points with centroid and accordingly assign points to the clusters.

Step 3: Now with the points in each cluster calculate the centroid by taking average of points of the cluster.

Step 4: Again, follow step 2 and 3 of assigning points to the clusters and recalculating of centroids. Repeat process until there in not much change come in centroid.

As now we are getting data into desired number of groups(cluster).

So, by applying this algorithm to our dataset we are going to divide the data values into three groups namely low, medium and high clusters.

- Conditions of which value comes under low group(cluster) are not feasible condition as these conditions may harm the field and soil conditions.
- Conditions corresponding to medium range values are best suitable for working.
- Conditions of whose values comes under high group(cluster) are also not feasible as excess things may lead to unproper growth of crop.

Now as once we divide the data in three clusters, we also get the information of minimum values of each feature of data corresponding to each cluster and with help of those can say whether the properties of future conditions are feasible or not.

4.7.4 Working code:

(The code has been written in Google Colab)

<u>Data Source</u>: Data from sensors are going in ThingSpeak Server and then from that Server we are exporting the data file.

So, in our code feeds.csv is the name of the file which contains the data which is coming from sensors.

Data looks like:

A	B	C	D	E
created_at(Time(	Temperature	Humidity	Soil Moisture	
2021-03-16 19:31:17 U	29.7	37.00	99.9	
2021-03-16 19:31:38 U	29.7	38.00	99.92	
2021-03-16 19:31:59 U	30.4	53.00	99.23	
2021-03-16 19:33:11 U	29.9	40.00	99.244	
2021-03-16 19:33:27 U	29.8	40.00	98.44	
2021-03-16 19:34:09 U	29.7	39.00	98.56	
2021-03-16 19:34:26 U	29.7	39.00	98.66	
2021-03-16 19:34:42 U	29.6	39.00	98.43	
2021-03-16 19:34:58 U	29.6	39.00	98.42	
2021-03-16 19:35:14 U	29.7	39.00	98.1	
2021-03-16 19:35:30 U	29.7	39.00	95.33	
2021-03-16 19:35:50 U	29.7	39.00	94.33	
2021-03-16 19:36:08 U	29.7	39.00	64.3	
2021-03-16 19:36:24 U	29.8	39.00	68.3	
2021-03-16 19:36:41 U	29.8	39.00	56.99	
2021-03-16 19:36:59 U	29.8	39.00	56.55	
2021-03-16 19:37:16 U	29.8	39.00	8	
2021-03-16 19:37:33 U	29.9	39.00	6.52	
2021-03-16 19:37:49 U	29.9	38.00	4.32	
2021-03-16 19:38:05 U	29.9	38.00	2	
2021-03-16 19:38:21 U	29.9	38.00	0	
2021-03-16 19:38:38 U	29.9	38.00	0	

feeds

Figure 35 Data Set

+ Code + Text

```
[2] import pandas as pd
    import numpy as np
    from sklearn.preprocessing import MinMaxScaler
    from matplotlib import pyplot as plt

    %matplotlib inline
```

```
[3] from google.colab import drive
    drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
[4] df = pd.read_csv("/feeds.csv")
    print(df)
```

```
                  created_at  Temperature  Humidity
0    2021-03-16 19:31:17 UTC         29.7      37.0
1    2021-03-16 19:31:38 UTC         29.7      38.0
2    2021-03-16 19:31:59 UTC         30.4      53.0
3    2021-03-16 19:33:11 UTC         29.9      40.0
4    2021-03-16 19:33:27 UTC         29.8      40.0
..                       ...          ...       ...
540  2021-03-17 19:07:19 UTC         30.6      34.0
541  2021-03-17 19:07:36 UTC         30.6      34.0
542  2021-03-18 10:16:42 UTC         29.5      44.0
543  2021-03-18 10:17:35 UTC         29.6      43.0
544  2021-03-18 10:22:23 UTC         29.6      44.0

[545 rows x 3 columns]
```

```
plt.scatter(df.Temperature,df['Humidity'])
plt.xlabel('Temperature')
plt.ylabel('Humidity')
```

Figure 36 Import and read data in code

\+ Code + Text

```python
scaler = MinMaxScaler()

scaler.fit(df[['Humidity']])
df['Humidity'] = scaler.transform(df[['Humidity']])

scaler.fit(df[['Temperature']])
df['Temperature'] = scaler.transform(df[['Temperature']])
```

```
[7] print(df)
```

```
                  created_at  Temperature  Humidity
0    2021-03-16 19:31:17 UTC     0.055556  0.392857
1    2021-03-16 19:31:38 UTC     0.055556  0.428571
2    2021-03-16 19:31:59 UTC     0.250000  0.964286
3    2021-03-16 19:33:11 UTC     0.111111  0.500000
4    2021-03-16 19:33:27 UTC     0.083333  0.500000
..                       ...          ...       ...
540  2021-03-17 19:07:19 UTC     0.305556  0.285714
541  2021-03-17 19:07:36 UTC     0.305556  0.285714
542  2021-03-18 10:16:42 UTC     0.000000  0.642857
543  2021-03-18 10:17:35 UTC     0.027778  0.607143
544  2021-03-18 10:22:23 UTC     0.027778  0.642857

[545 rows x 3 columns]
```

```
[33] km = KMeans(n_clusters=3,init='k-means++',n_init=10, max_iter=300,tol=0.0001)
     y_predicted = km.fit_predict(df[['Temperature','Humidity']])
     y_predicted
```

Figure 38 Defining value of K (No. of Clusters)

\+ Code + Text

```
[33] array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
            1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
            1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
            1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
            1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
            1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
            1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
            1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
            2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
            2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
            0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
            0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
            0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
            0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
            0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
            0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
            0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
            2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
            2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
            2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
            2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
            2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
            2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
            2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
            2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1], dtype=int32)
```

Figure 37 Grouping of data into clusters

```
[37] df['cluster']=y_predicted
     print(df)
```

```
                   created_at  Temperature  Humidity  cluster
0    2021-03-16 19:31:17 UTC     0.055556  0.392857        1
1    2021-03-16 19:31:38 UTC     0.055556  0.428571        1
2    2021-03-16 19:31:59 UTC     0.250000  0.964286        1
3    2021-03-16 19:33:11 UTC     0.111111  0.500000        1
4    2021-03-16 19:33:27 UTC     0.083333  0.500000        1
..                       ...          ...       ...      ...
540  2021-03-17 19:07:19 UTC     0.305556  0.285714        2
541  2021-03-17 19:07:36 UTC     0.305556  0.285714        2
542  2021-03-18 10:16:42 UTC     0.000000  0.642857        1
543  2021-03-18 10:17:35 UTC     0.027778  0.607143        1
544  2021-03-18 10:22:23 UTC     0.027778  0.642857        1

[545 rows x 4 columns]
```

Figure 39 Assigning cluster to data sheet

```
+ Code  + Text

[34] km.cluster_centers_

     array([[0.8956003 , 0.0738255 ],
            [0.11354167, 0.43928571],
            [0.42890772, 0.25015133]])

[35] df1 = df[df.cluster==0]
     df2 = df[df.cluster==1]
     df3 = df[df.cluster==2]
     plt.scatter(df1.Temperature,df1['Humidity'],color='green')
     plt.scatter(df2.Temperature,df2['Humidity'],color='red')
     plt.scatter(df3.Temperature,df3['Humidity'],color='black')
     plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color='purple',marker='*',label='centroid')
     plt.legend()
```

Figure 40 Colour to each Cluster

4.7.5 Output From code:

```
<matplotlib.legend.Legend at 0x7f730e712390>
```

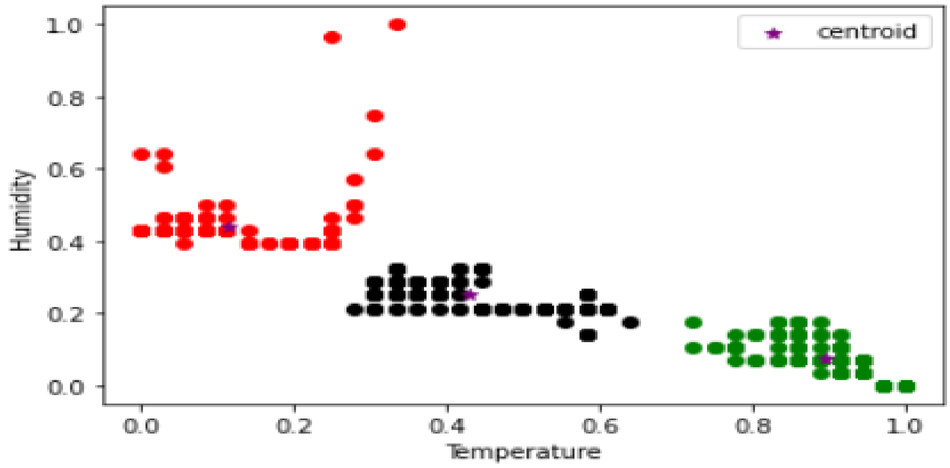

Figure 41 Clustering of data of Temp. and Humidity in red, black and green colour

Similarly, we have applied this algorithm to temperature and soil moisture values and we get the result as: -

```
<matplotlib.legend.Legend at 0x7f0b2b782950>
```

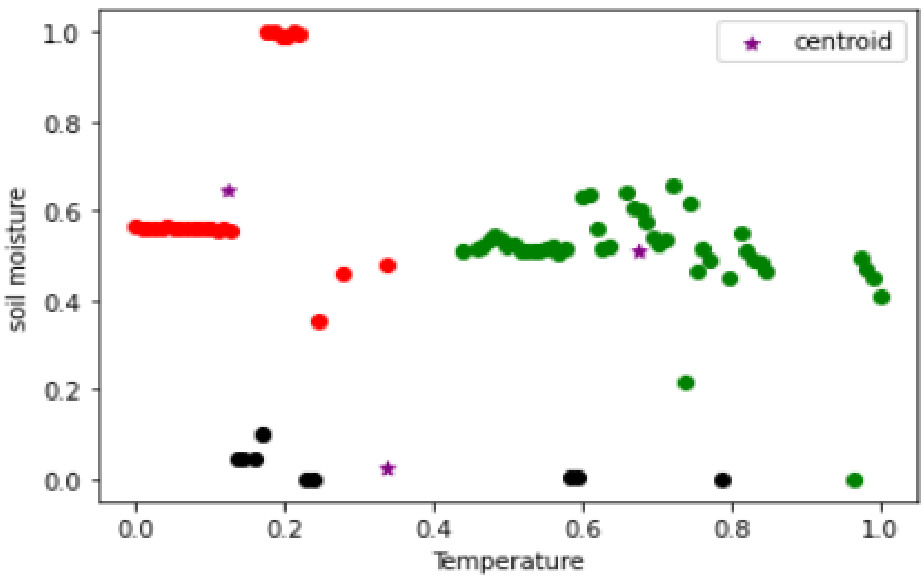

Figure 42 Clustering of Temperature and soil moisture data values

4.7.6 Keynotes from code:

- As in our output graph we can see that we have formed three clusters, and with the help of this code we can find we can find minimum / threshold data value for each cluster.
- And can predict the conditions with the future value by comparing it will each cluster's threshold value.
- As we have normalised our data so from output graph, we can see that some data is in low range some in medium and some in high. So, this will help in knowing soil conditions for future references.
- Therefore, by plotting at different time interval we are able to see conditions and then can react accordingly.

CHAPTER 5

CONCLUSION

5.1 Conclusion

- In this project, a real-time data collection of soil parameters, namely temperature, humidity and soil moisture level, has been established through a microcontroller driven circuit.
- A CAD model of the actual A-IoT Cube has been created in AutoCAD Software, inside which the circuit will be mounted.
- Communication establishment between circuit and the Cloud platform has been implemented with the help of an ESP8266 Wifi Module, that transfers data to the cloud platform 'ThingSpeak' with minimal Internet usage.
- A Machine Learning Algorithm Model has been established which uses K-Means Clustering Algorithm to determine various soil parameter limits, and will be further more accurate when tested on a real farm.
- A Mobile Application has been developed on the MIT App Inventor Platform which shows the sensor readings in a digital and graphical format in real-time with again, a minimal Internet usage.
- An alerting system has been set up that sends an alert message to a defined Mobile number and sends an Email alert to a defined Email ID whenever any soil parameter crosses a safe limit, in a cost-free way, using the platform IFTTT for the same.

5.2 Future Scope

The project in future, aims the model to be developed to a better extent in the ways of testing the model on a real farm and training the ML model on a variety of crops to have the determined limits for a greater number of crops and hence increase a parameter of setting in the model where the farmer mentions the crop he is going for and accordingly every connected system adapts. This has been a limiting factor till now for this project considering the lockdown situation that has barred the model to be tested on a real farm. A higher degree of digitalisation can be introduced by connecting irrigation facility to this model, which can be operated by a click on the app.

CHAPTER 6

REFERENCES

6.1 References

- Shi, X.; An, X.; Zhao, Q.; Liu, H.; Xia, L.; Sun, X.; Guo, Y. State-of-the-Art Internet of Things in Protected Agriculture. Sensors 2019, 19, 1833.

- M. Ayaz, M. Ammad-Uddin, Z. Sharif, A. Mansour and E. M. Aggoune, "Internet-of-Things (IoT)-Based Smart Agriculture: Toward Making the Fields Talk," in IEEE Access, vol. 7, pp. 129551-129583, 2019.doi: 10.1109/ACCESS.2019.2932609

- Barun Basnet, Junho Bang, "The State-of-the-Art of Knowledge-Intensive Agriculture: A Review on Applied Sensing Systems and Data Analytics", Journal of Sensors, vol. 2018, Article
ID 3528296, 13 pages, 2018. https://doi.org/10.1155/2018/3528296

- Website references:

  - https://thingspeak.com/pages/commercial_learn_more?s_tid=SmartAgriculture

  - http://eagri.org/eagri50/AGRO301/index.html

  - https://www.nrcs.usda.gov/Internet/FSE_DOCUMENTS/nrcs142p2_053280.pdf

  - https://appinventor.mit.edu/explore/ai2/tutorials

  - https://realpython.com/k-means-clustering-python/