A PROJECT UNDER SWMLIT 2018

ON

"INTELLIGENT GARDENING SYSTEM"

Submitted to

Indian Institute of Technology (BHU)

BY

Ram Mahesh 126

Abhishek Kumar 127

UNDER THE GUIDANCE OF

Prof. Hari Prabhat Gupta

DEPARTMENT OF COMPUTER SCIENCE

Indian Institute of Technology (BHU)

Varanasi, Uttar Pradesh, India

A PROJECT REPORT
ON

"INTELLIGENT GARDENING SYSTEM"

Submitted to

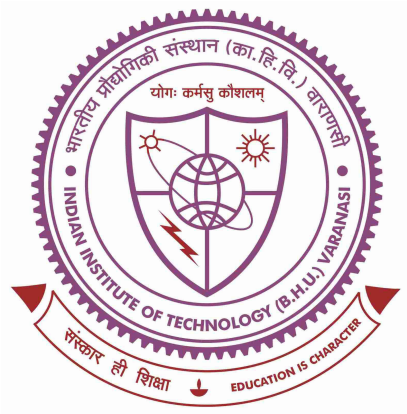
Indian Institute of Technology (BHU)

BY

Ram Mahesh	**126**
Abhishek Kumar	**127**

UNDER THE GUIDANCE OF
Prof. Hari Prabhat Gupta

DEPARTMENT OF COMPUTER SCIENCE
IIT(BHU)



India Institute of Technology (BHU)

Acknowledgements

We are profoundly grateful to **Prof. Hari Prabhat Gupta** for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion.

At last we express our sincere heartfelt gratitude to **Rahul Mishra** research scholar Computer Science and Engineering Department who helped me directly or indirectly during this course of work.

RAM MAHESH
ABHISHEK KUMAR

ABSTRACT

Abstarct : With the rise in the demand of automation in industries and agriculture, we thrive for well optimized automatic solutions, which require least human interventions. The project idea is to achieve this in the field of agriculture. This low energy system is to give the automation to existing irrigation systems an intelligence with the help of machine learning. Raspberry pi along with all the sensors detects the current surrounding conditions like temperature, humidity, moisture, wind speed and the processed image of plant, and applies the K-Nearest Neighbor algorithm on the dataset recorded by us previously. A user-friendly GUI also runs parallelly on the raspberry pi local server, through which one can monitor the whole process. The overall is a low cost system, which caters to the need of various agricultural industries and smart farming techniques.

Keywords: automation in agriculture,Machine Learning Applications in Agriculture irrigation, Smart agriculture, IoT agriculture,

Contents

List of Figures

ABSTRACT

Abstarct : With the rise in the demand of automation in industries and agriculture, we thrive for well optimized automatic solutions, which require least human interventions. The project idea is to achieve this in the field of agriculture. This low energy system is to give the automation to existing irrigation systems an intelligence with the help of machine learning. Raspberry pi along with all the sensors detects the current surrounding conditions like temperature, humidity, moisture, wind speed and the processed image of plant, and applies the K-Nearest Neighbor algorithm on the dataset recorded by us previously. A user-friendly GUI also runs parallelly on the raspberry pi local server, through which one can monitor the whole process. The overall is a low cost system, which caters to the need of various agricultural industries and smart farming techniques.

Keywords: automation in agriculture,Machine Learning Applications in Agriculture irrigation, Smart agriculture, IoT agriculture,

Chapter 1

Introduction

In today's world when the human population is growing rapidly and thus rising demand of agricultural produce.Our farmers need to go field to decide when to water their fields.Future generation which lacks interest in the farming soon there going to be lack of manpower in the agriculture.
Certainly technology is only way that could play an important role in improving declining condition of agriculture.
This project is aimed at developing an autonomous system capable of taking decision when to water a plant? This project will be a lot helful for farmers.When one needs to irrigate a large field instead of visiting each and every nook of such big field he can rely over this system.

Figure 1.1: Traditional Irrigation System

1.1 Problem Statement

To design a system that is able to take decision when to water plants based on the condition of environment sensed by the various sensors.

1.2 Current Scenario

In the present time, the use of sensors are limited to take the readings and make decisions if a certain condition mets. For a more complex application, we need to write down or define a lot of conditions to make our system take decisions. There is no intelligence in them.They are hard-coded and it's troublesome to manage the multiple variables and to make condiitons based upon them.Currently in market there is no such product to automatate the irrigation system.

1.3 Our Approach

We have tried to implement the intelligence in the watering systems, in which there would be no human interventions. The machine will take decision by it's own depending upon its past experiences.We had implemented it using different machine learning approaches & models to predict weather there is need to water the plant or not. We had tried to this project in a way that if needed we could make commercial product based upon this.

Chapter 2

Project Setup

For an IOT system the creating a sensor node is much important part. The values from the environment of the plant are sensed and converted to electrical signals by the help of sensors.The decision taken by the microcontroller is being converted in the useful form only when the data is converted to another form of energy by the help of actuators (Motor,Relay)

2.1 Sensors

2.1.1 Moisture & Humidity Sensors (DHT11)

To sense the temperature and humidity of the surrounding of the plant we had used the DHT11 sensor.
DHT consist of a humidity sensing component, a NTC temperature sensor (or thermistor) and an IC on the back side of the sensor.
For measuring humidity it uses the humidity sensing component which has two electrodes with moisture holding substrate between them. So as the humidity changes, the conductivity of the substrate changes or the resistance between these electrodes changes. This change in resistance is measured and processed by the IC which makes it ready to be read by a microcontroller.
To measure temperature these sensors use a NTC temperature sensor or a thermistor.A thermistor is a variable resistor that changes its resistance with change of the temperature.
To take the readings of this sensor we had used the Adafruit library of the DHT11 and connected DHT11 to the second GPIO pin of the Raspberry. The library used handles the task like as conversion to appropriate format and value.

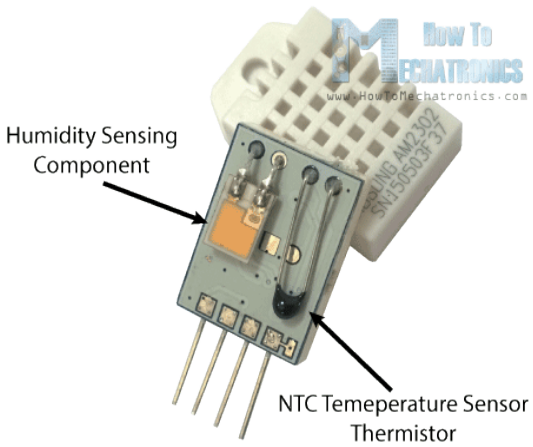


Figure 2.1: Components of DHT11

2.1.2 Rain Sensor

Raindrop sensor is basically a board on which nickel is coated in the form of lines. It works on the principal of resistance.
Rain Sensor module allows to measure moisture via analog output pins and it provides a digital output when a threshold of moisture exceeds.
The module is based on the LM393 op amp. It includes the electronics module and a printed circuit board that collects the rain drops. As rain drops are collected on the circuit board, they create paths of parallel resistance that are measured via the op amp. The same signal is then send to the digital GPIO pin of Raspberry Pi (its GPIO pin 5 in our case).

Figure 2.2: Rain Sensor Module

2.1.3 Pi Camera

Raspberry Pi camera is an important part of this project as it enables to take the images of plant. These images are later on processed using the OpenCV to find the

amount of green colour in plant.

The camera connects to the Raspberry Pi's Camera Serial Interface (CSI) bus connector via a flexible ribbon cable. The camera's is of five megapixels and has a fixed focus lens. The software for the camera supports full resolution still images up to 2592x1944 and video resolutions of 1080p30, 720p60 and 640x480p60/90. The camera module is shown below:

Figure 2.3: Raspberry Pi Camera

2.1.4 Soil Moisture Sensor

We have used soil moisture sensor FC-28 to record the moisture values of soil. The sensor is equipped with both analog and digital output, so it can be used in both analog and digital mode. The soil moisture sensor consists of two probes which are used to measure the volumetric content of water. The two probes allow the current to pass through the soil and then it gets the resistance value to measure the moisture value.
When there is more water, the soil will conduct more electricity which means that there will be less resistance. Therefore, the moisture level will be higher. Dry soil conducts electricity poorly, so when there will be less water, then the soil will conduct less electricity which means that there will be more resistance. Therefore, the moisture level will be lower.
It takes the input voltage of 3.3V to 5V and draws 35mA of current. It outputs upto 0-4.2 Volts. We have taken the analog values from the sensor by reading floating values from GPIO pin.

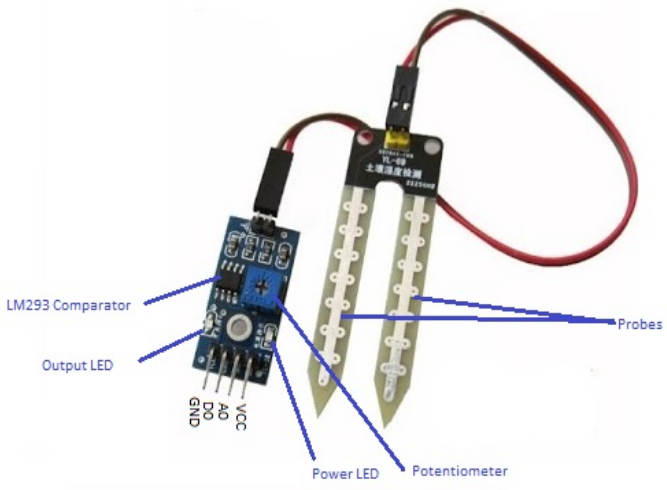


Figure 2.4: Soil moisture Sensor Module

2.2 Actuators

2.2.1 Relay

These are electromagnetic switches,a relay switch has one or more poles, each of whose contacts can be thrown by energizing the coil. Normally open (NO) contacts connect the circuit when the relay is activated; the circuit is disconnected when the relay is inactive.These are being operated by using the low DC voltage(12V) and are used to control the on and off operation of the electrical devices like as the motor.

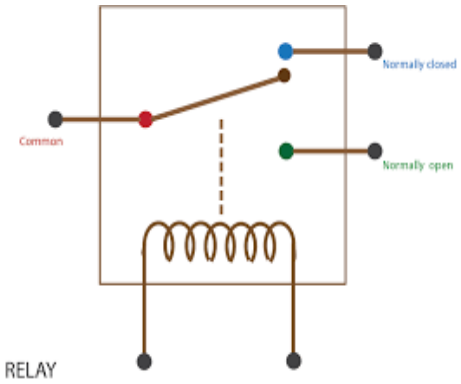


Figure 2.5: Relay Internal Diagram

2.2.2 Buzzer

Buzzer is used to give user an indication that the data is being recorded by the Raspberry Pi.Each time the buzzer beeps the data is being written to the csv file.

Figure 2.6: Buzzer

2.2.3 Electric Motor

2.3 Controlling System

2.3.1 Arduino

This is a microcontroller that has been used to collect the valus from the moisture sensor and to control the operation of the relay.
Raspberry Pi being unable to read the analog values from sensors and to control the actuators requiring the high current it safe to control them from arduino.Thus we had used the arduino in our project here the Raspberry PI is a main controlling unit and the arduino is auxliary unit. [4]It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button.

2.3.2 Raspberry Pi

This is the heart of the whole project this is responsible for the all data collection and controlling task.
Various sensors and actuators have been attached to it which are controlled using this.The data preprocessing techniques and the machine learning algorithm have been implemented on this. The predictions for relay on/off status based upon the availble value from sensor is made using raspberry pi. Raspberry PI is a SOC (system on chip) it has following features.

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board

- 40-pin extended GPIO
- 4 USB 2 ports
- Full size HDMI
- CSI camera port for connecting a Raspberry Pi camera
- Micro SD port for loading your operating system and storing data

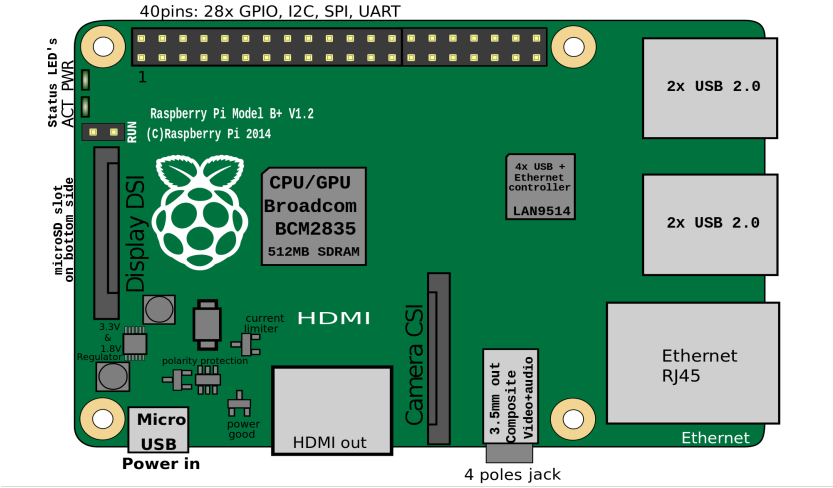


Figure 2.7: Raspberry PI-3

2.4 Project Implementation

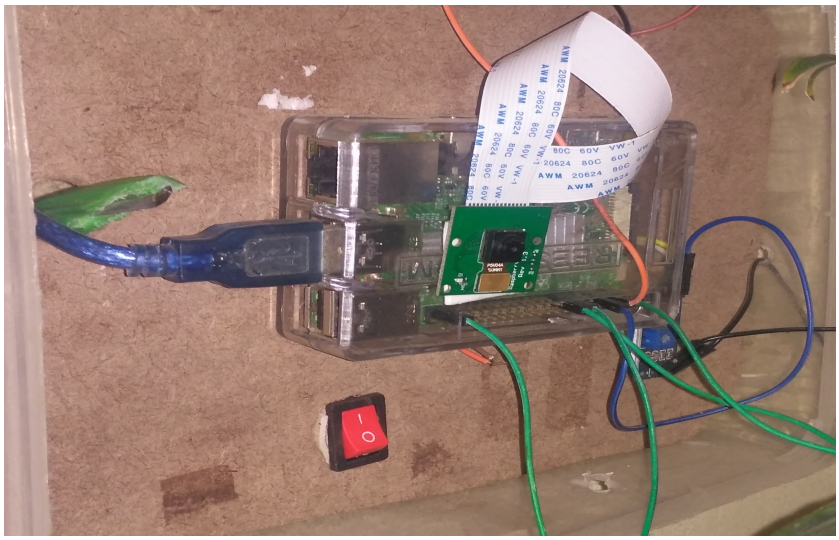


Figure 2.8: Hardware Setup

Chapter 3

Node Creation Setup

In order to extend the project to large scale assume over a large area over a filed we had tried to implement this using various data logging nodes and the control node which is responsible the overall functioning of the system we had tried to implement this project to cover a wide area usign multiple nodes.

3.1 Nodes Creation (Data logging and the Controlling Node)

Below digram shows the various Nodes involved in the project and their usage two nodes are data recording nodes while one is controlling node details about each node are mentioned in this chapter.

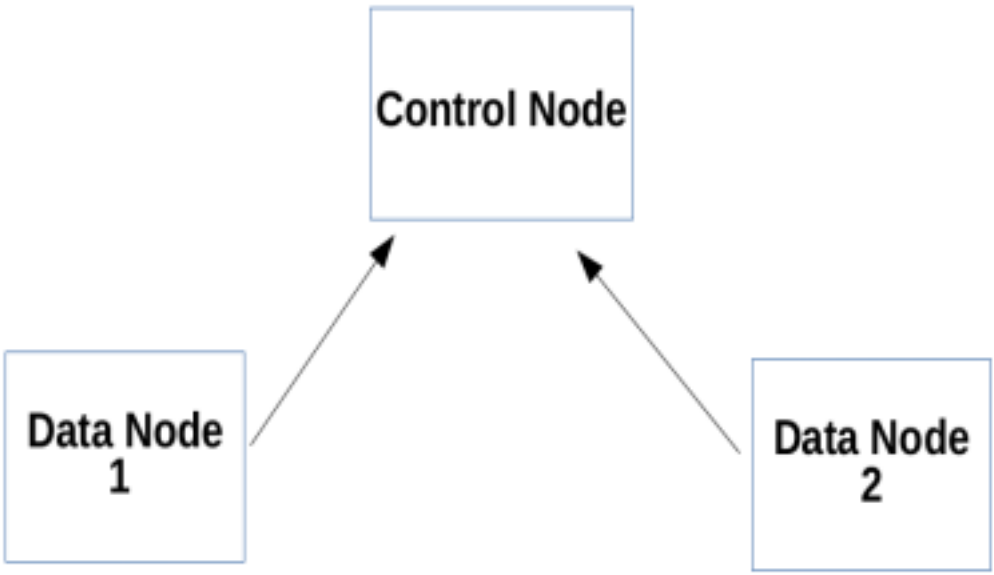


Figure 3.1: Nodes Involved in Project

3.2 Main Node (Controlling Node)

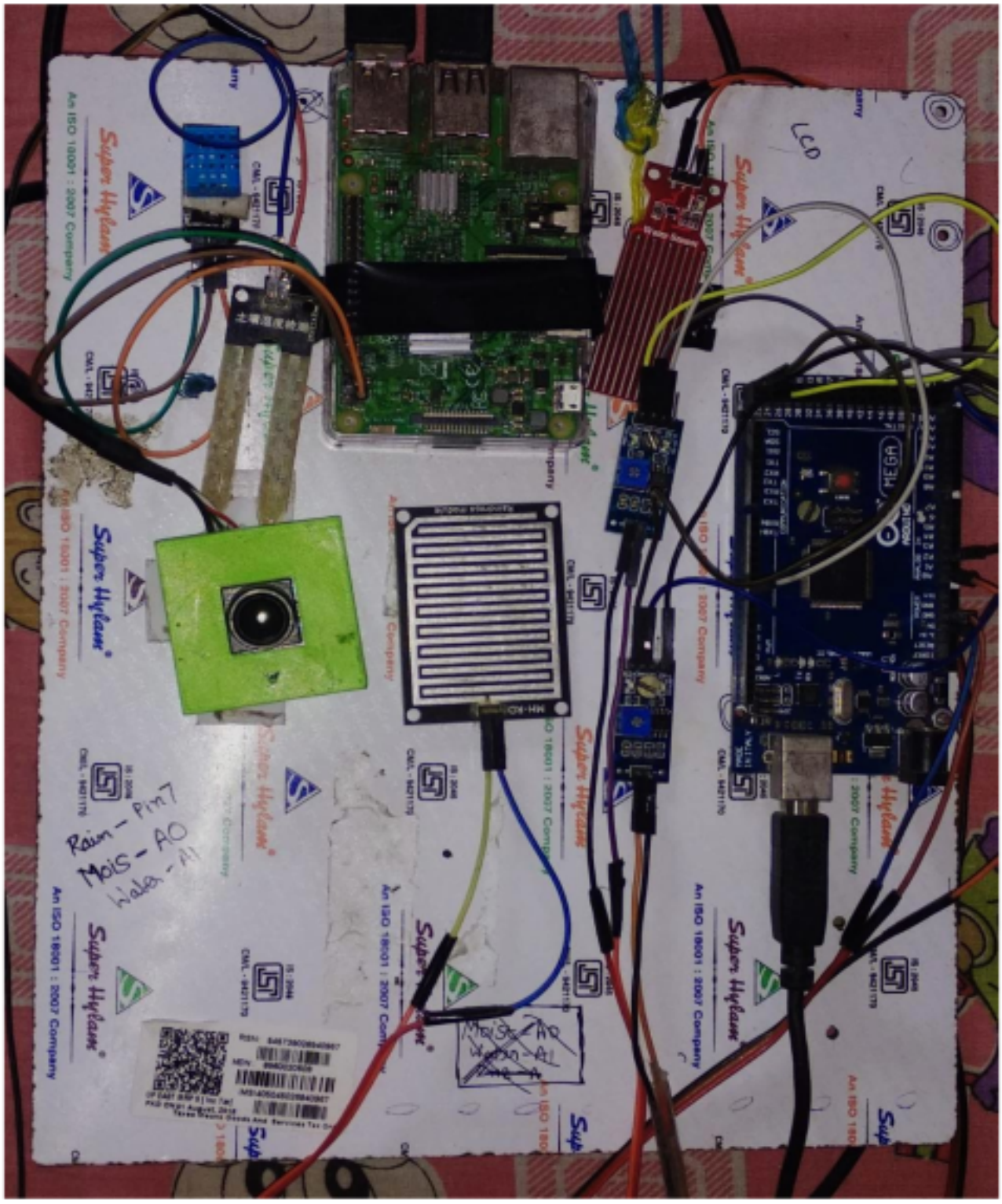


Figure 3.2: Control Node: This takes decisions regarding the switching of Relays and the Machine Learning Code Is running on this Raspberry Pi

3.2.1 Setup

Various Components attached to this node and their functioning is as mentioned be low.

- Webcam : This is used to take picture of the plant and to extract information about amount of the green colour present in captured plant image.
- Sensor : Sensors like as water level, moisture sensor are used to measure the various parameters of the plant while sensors like DHT11 and rain sensors are used to mesure environmental conditions.
- Indicator Leds: These are used to indicate the status of the relays that represent the on and off position of the water pump.

3.2.2 Functioning

Various tasks performed by this node are as follow.

- This node receives the values from rest of nodes and then stores them in a seperate line in the common data file. used to mesure environmental conditions.
- This Node takes the decision whether to switch ON/OFF the water pump based upon the values received from the data logging nodes. used to mesure environmental conditions.
- This node notifies the user about the status of the water pumps attached to remaining nodes.
- This node collects the information regarding environment parameters like as the temperature,humidity and the rain status

3.3 First Auxilliary Node (Data Logging Node)

Figure 3.3: Data Logging Node: This collects the information about the plant parameters like as the moisture, water level green content present in the each plant.

3.3.1 Setup

Various Components attached to this node and their functioning is as mentioned be low.

- Webcam : This is used to take picture of the plant and to extract information about amount of the green colour present in captured plant image.
- Sensor : Sensors like as water level, moisture sensor are used to measure the parameters of the plant. used to mesure environmental conditions.
- Raspberry Pi: This plays an important role this collects values from the sensors attached to the node and captures image of plant then calaculates the amount of the green colour present in the pant image used to mesure environmental conditions.
- Indicator Leds: These are used to indicate the status of the relays that represent the on and off position of the water pump.

3.3.2 Functioning

Various tasks performed by this node are as follow.

- Collects sensor values runs openCV values to collect the amount of greenary prestn in the plant.
- Sends collected values to the Control Node via MQTT.
- Receives command from the Control Node and switches ON/OFF the relays accordingly.

3.4 Second Auxilliary Node (Data Logging Node)

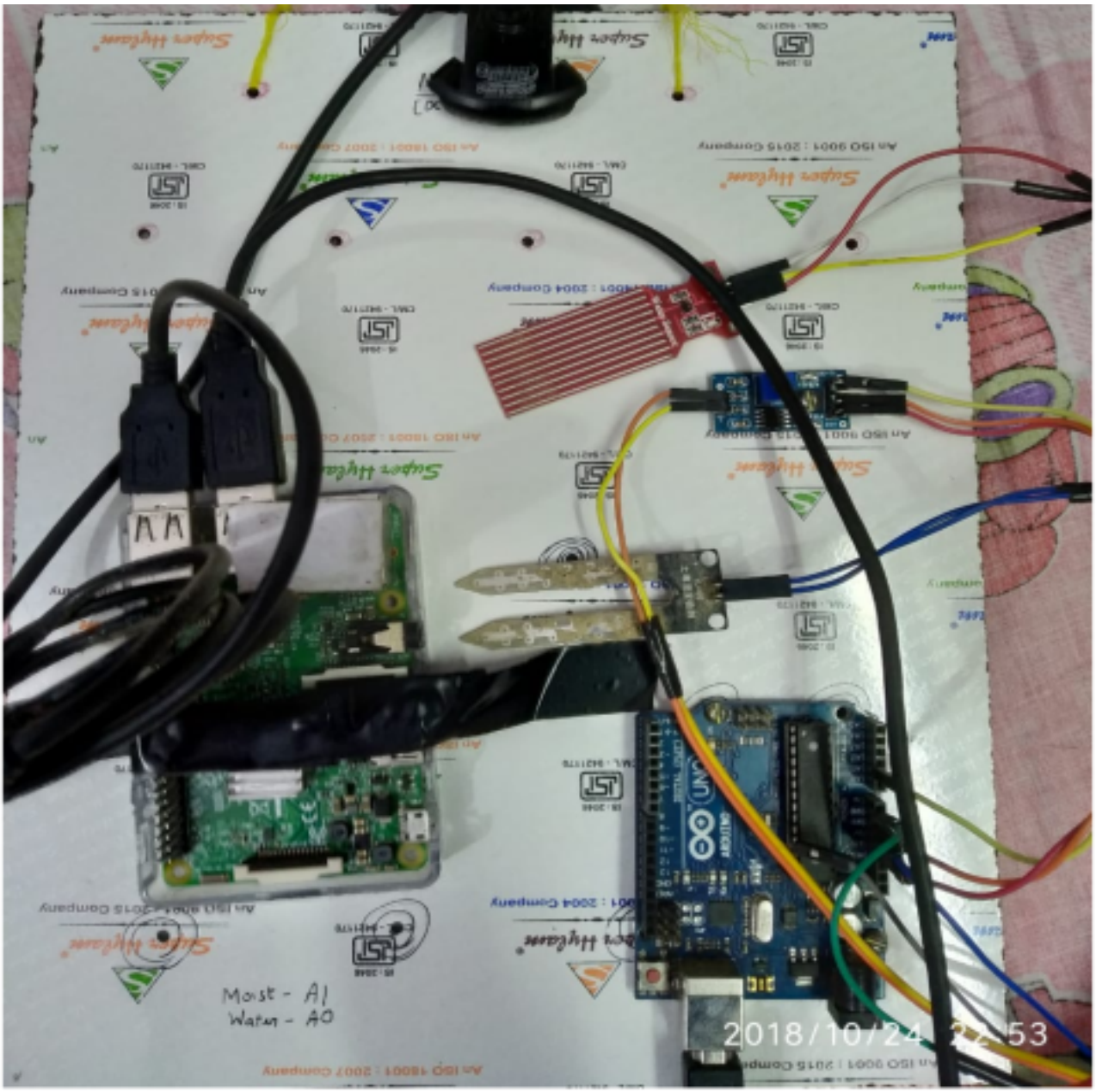


Figure 3.4: Data Logging Node: This collects the information about the plant parameters like as the moisture, water level green content present in the each plant.

3.4.1 Setup

Various Components attached to this node and their functioning is as mentioned be low.

- Webcam : This is used to take picture of the plant and to extract information about amount of the green colour present in captured plant image.
- Sensor : Sensors like as water level, moisture sensor are used to measure the parameters of the plant. used to mesure environmental conditions.

- Raspberry Pi: This plays an important role this collects values from the sensors attached to the node and captures image of plant then calaculates the amount of the green colour present in the pant image used to mesure environmental conditions.
- Indicator Leds: These are used to indicate the status of the relays that represent the on and off position of the water pump.

3.4.2 Functioning

Various tasks performed by this node are as follow.

- Collects sensor values runs openCV values to collect the amount of greenary prestn in the plant.
- Sends collected values to the Control Node via MQTT.
- Receives command from the Control Node and switches ON/OFF the relays accordingly.

Chapter 4

Networking Communication

Our project now involves the two data collection and one control node its necessarey to esatblish a link between them to transfer information from one node to the other node.We had implemented the Pi to Pi communication and the Pi - NodeMCU communication to acheive the above results we had used the MQTT algorithm.Using the MQTT algorithm the communicate among them selves.

4.1 PI - PI Communication

In order to make the necessary coordination among various nodes, we have deployed MQTT (Message Queuing Telemetary Transport) Protocol.

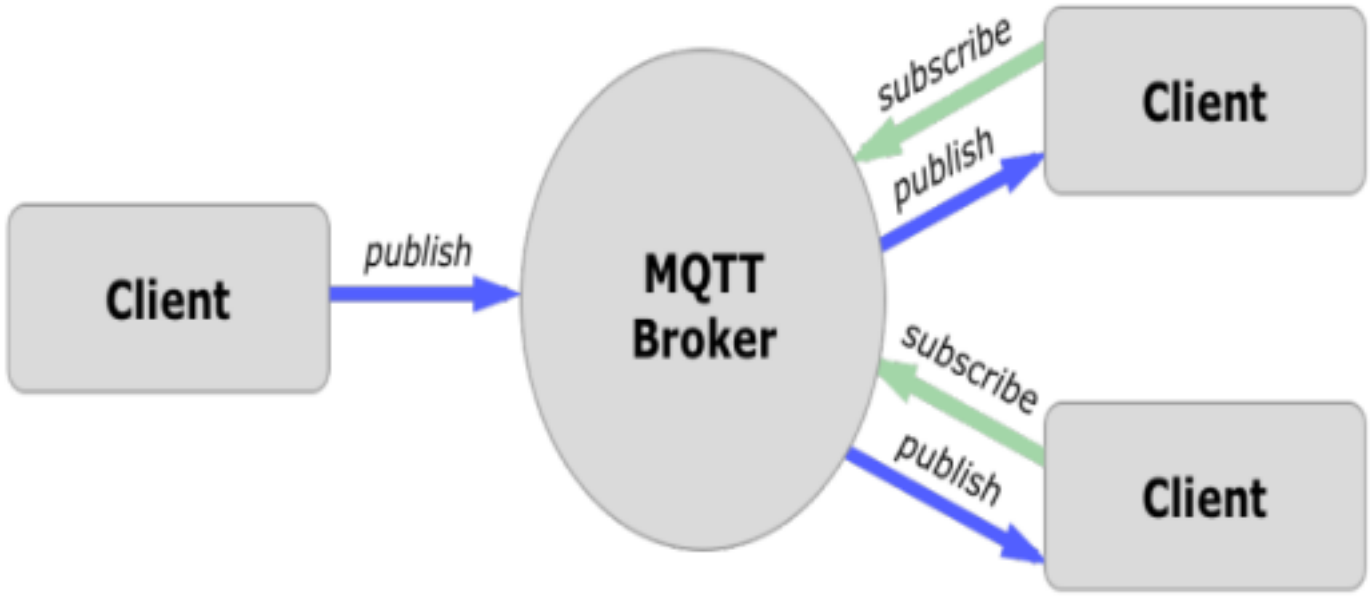


Figure 4.1: Basic Diagram of MQTT

4.1.1 Why MQTT?

MQTT is a very light weight protocol based on publish/subscribe method. Sender node behaves as publisher and recieving node behaves as subscriber.Publisher publishes the the message on a topic with unique name, and that topic is subscribed by the subscriber in order to recieve the message. Since, we ocassionaly require to send floating values of humidity and water level, this protocol is best suited among all other protocols.//

4.1.2 Implementation

In order to implement this on Raspberry Pi-3, we made two files, named mqtt-publish.py mqtt-subscribe.py The mqtt-publish.py is executed on sending nodes and mqtt-subscribe.py executed on main controlling node. These files are saved on Raspberry pi of controller node and sub nodes. Libraries for mqtt were imported as.

```
import paho.mqtt.client as mqtt
import paho.mqtt.publish as publish
```

Figure 4.2: Libraries

After writing the code and then we ran the code using the terminal we got the following reponse which shows that the two PI are connected to each other.

```
  File "/usr/local/lib/python2.7/dist-packages/
    socklist = select.select(rlist, wlist, [],
KeyboardInterrupt
pi@raspberrypi:~$ python mqtt_subscriber.py
connected with result code 0
```

Figure 4.3: Output of MQTT subscriber

Now we had transferred a character from the one Pi to the other Pi. As shown below.

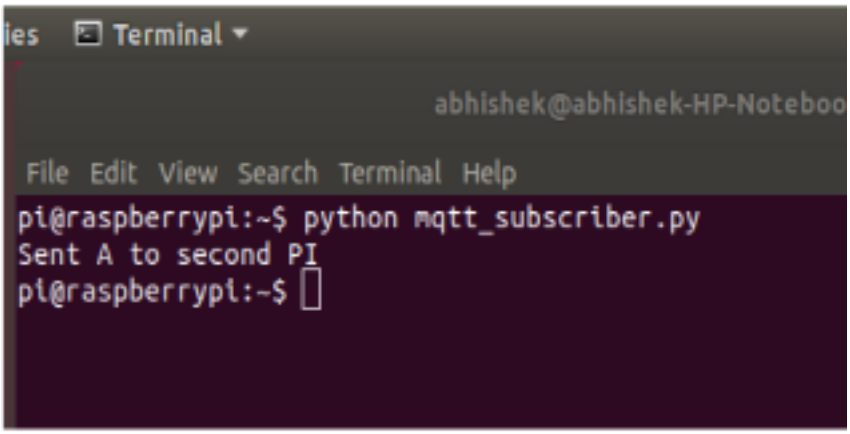


Figure 4.4: Data sent using MQTT

The same character has been recieved on the other raspberry Pi that is kept beside it and character gets transferred wirelessly.

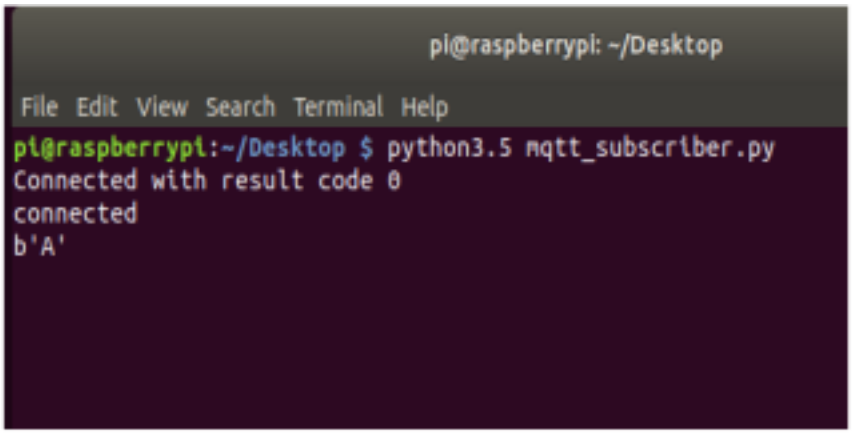


Figure 4.5: Data received using MQTT

4.1.3 Using MQTT in implementation of Setup

We have utilised the property that a node can behave as both publisher and subscriber simultaneously. Whenever main (or controlling) node needs values of different sensors from other nodes, it sends a unique character to the nodes. In reply, the nodes will send the comma separated values of sensors to the main node. Hence, it leads to proper synchronization among all the nodes. Thats how they communicate with each other.//

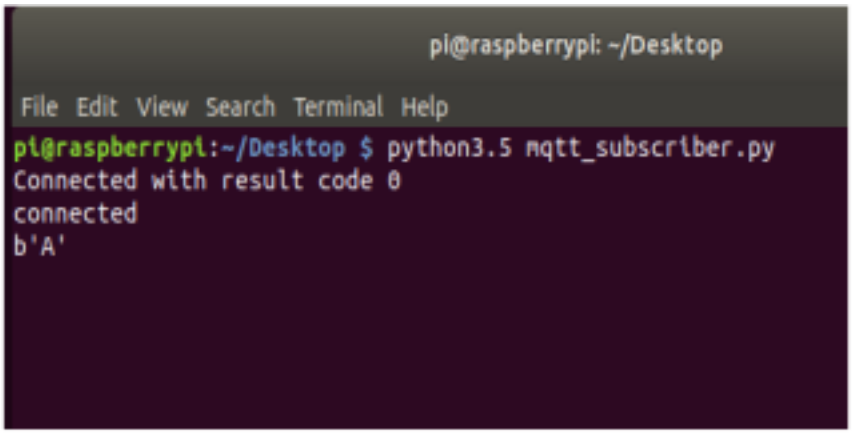


Figure 4.6: Data received using MQTT

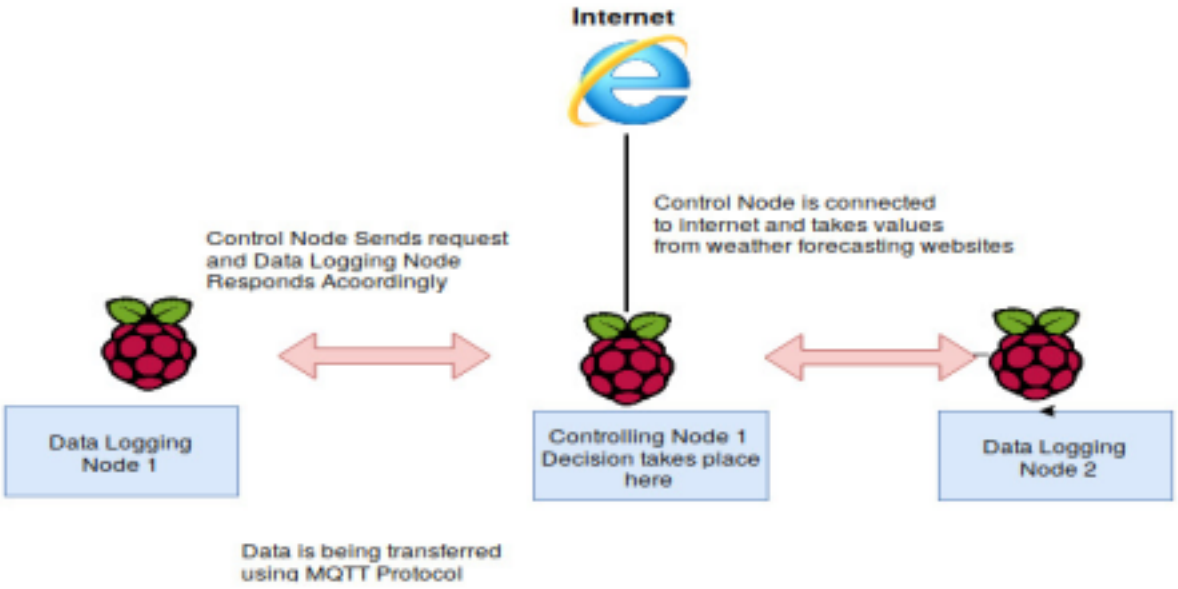


Figure 4.7: Communication Overview

We had used the MQTT protocol to establish the following request and subscribe model as shown below.In wireless sensor networks we need to take care of the power consumption,placement of the nodes in the geographical area. We follow the below mentioned steps to implement the project.

- Controller Pi sends request (character) to node-1 Pi.
- The arduino present on the node-1 recieves the request and it turn on the relay to switch on Data logging center 2 (or node 2).
- Now the both of the Data logging centres are active.
- The controller node sends a character to both data logging centres requesting the information

- After both the nodes had sent data to the Main Node both the Raspberry Pi are now shut down.
- After shutting down the relay is now turned off thus the node 2 gets switched off while node 1 remains active

4.2 Placement In Geographical Area

For proper placement of nodes in a feild or geogrophical area, we have followed the OGDC algorithm Optimal Geographical Density Control (OGDC) algorithm consider dense deployment for sensors in the monitoring area. Idea of this algo- rithm is shown in Fig. 4.7. The result shows the relationship between sensing range and communication range. It is also shown that to cover one crossing point of two circles, only one circle should be used and center of all these circles should form an equilateral triangle of side length 3 * R, where R is radius of circle. In this algorithm, three states for sensor nodes are defined undecided, on and off. Initially all sensor nodes are in undecided stare. Based on some threshold value, one node starts the activation round. Then another node at a distance equal to 3 times of sensing range is selected to become as active node. Similarly, third node from these 2 nodes is selected at a distance equal to 3 times of sensing range. In this way nodes are activated in each round by considering their distance from active nodes and their power level.

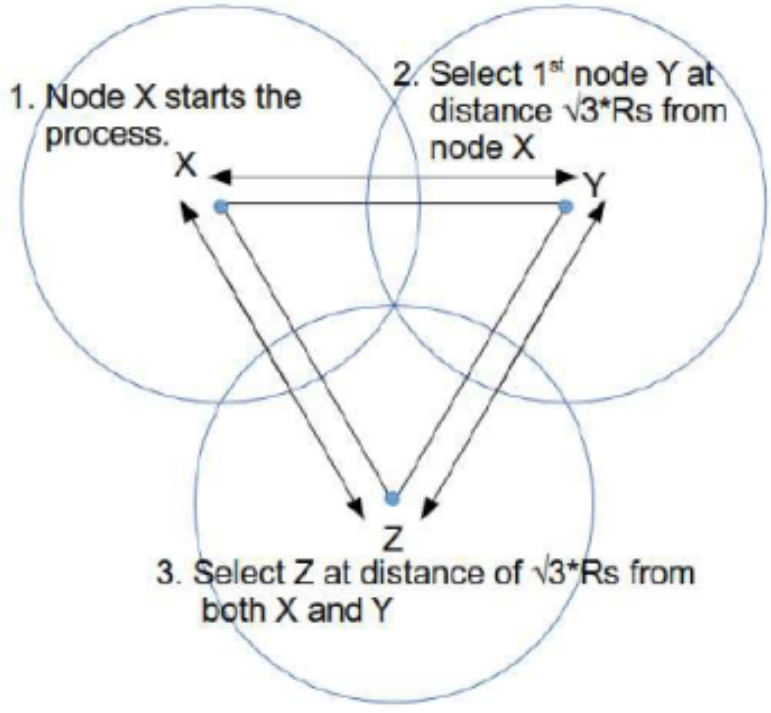


Figure 4.8: Optimum Geographical Density Control

From this for this particular node we have to ensure that there has to be another node somewhere which is going to collect that data that is transmitted by this particular node, at it receives it and through a multi hop path the data has to be transmitted to the sink node has to be related to the sink node or the base station. So, the few essential concepts, it has been shown in the literature that there is a relationship be- tween coverage and connectivity. So, it has been shown that if we ensure that the transmission range of a sensor node is greater than or equal to 2 times the sensing rangethen coverage would automatically imply connectivity. In sensor networks, providing full coverage is not sufficient. Established network should also be connected. Connected network means given any two sensor nodes, they are able to communicate directly or with the help of intermediate sensor nodes. Connected network also imply that sensed data through any sensor node can reach sink/base-station because final processing and data transmission to remote user is done by sink nodes. A lot of work has been done to provide full coverage along with network connectivity. The objective of the algorithm is to maximize the network lifetime or minimize the number of sensors that are deployed.

Chapter 5

Data Collection

In Machine learning projects the data plays an important role.It is an necessary task to collect the data from all the sensors and storing them correctly in a proper file format so that they can be used when nedded to design the various machine learning algorithms on the project. We had collected the Following data from the surrounding of the plant and stored them in comma seperated file.

- Temperature
- Humidity
- Status wheather raining or not.
- Soil Moisture
- Amount of green colour present in the Image captured by Camera
- Wind Speed (From the Internet)
- Status of relay it should be switched off or not
- Date and Time
- Node Number
- Water Level

Figure 5.1: Dataset

In this chapter we will descibe the ways by which we had collected sensor reading and how we stored them in a file.

5.1 Sensors Data Collection

5.1.1 Temperature & Humidity Data Collection

We had used the DHT11 for the collection of the temperature and humidity data from the surrounding of the plant for this purpose we had used the **Adafruit_DHT** library and stored it's output in the file.This library has a Adafruit_DHT.read_retry(11,2),this command fetches the sensor values from the pin number 2 of Raspberry Pi to which sensor is being connected.Shown below is the image of the output when this python command is being executed in the terminal window of the UBUNTU.

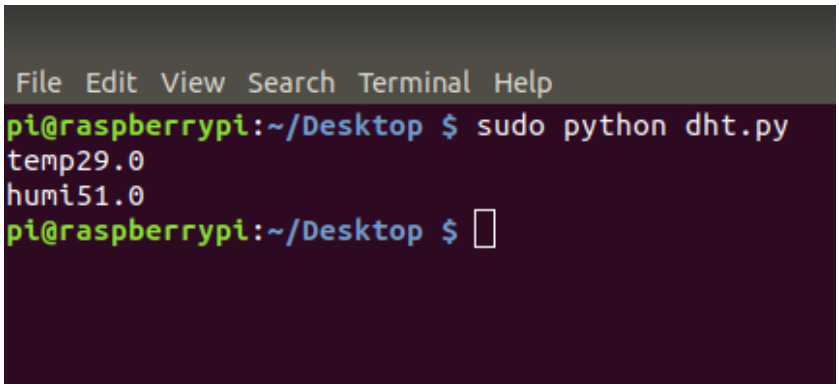


Figure 5.2: Temperature from DHT11

5.1.2 Rain Status

To know wheather it is raining or not we had used the Rain Sensor. Which provides output as 1 if it is not raining and the 0 if it is raining.This sensor provides the output in the form of the High or Low depending upon the weather conditions.We had used the GPIO pins of Raspberry Pi to read the sensor value.

5.1.3 Soil Moisture and Water Level Data Collection

To calculate the soil moisture and Water Level content we had to record the analog valus coming from the plant moisture sensor. Since the Raspberry Pi lacks the ability to record the analog values as it doesn't has any Analog to Digital Converter. To overcome this problem we had used the arduino and send data from it to Raspberry Pi by using as a serial interface.We had connected arduino to Raspberry Pi through USB Port. Serial communications are essential for every Micro-controllers to communicate between Micro-controllers and another device. The Micro-controller sends these 1 and 0 (bits) that contain necessary information one by one, or Serially.WE had tranmitted values of the Soil Moisure sensor from arduino to Raspberry Pi over serial communication.

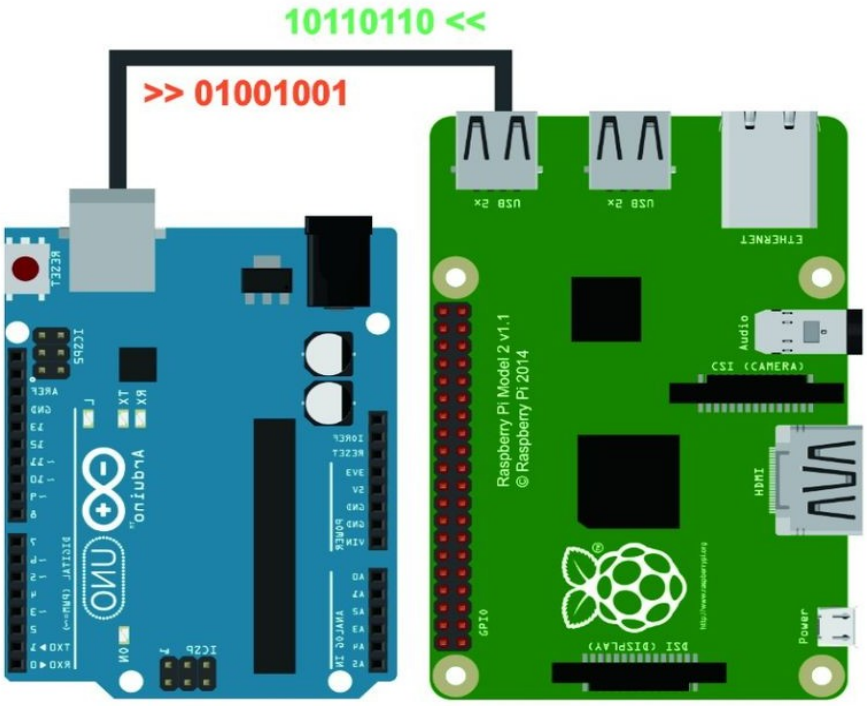


Figure 5.3: Connection of Raspberry & Arduino

5.1.4 Time Stamp

To identify the data easily and to track the time and date on which data was being recorded. To implement this we had used python library to fetch the date and time. This library synchronizes the time and date of the Raspberry Pi to the Internet since the raspberry pi lacks the Real Time Clock.So everytime we shud down the raspberry pi it's time and date changes.To overcome this problem we had used the python library to overcome this.

5.1.5 Wind Speed Collection

To record the Wind Speed of the environment we had imported the data from the **www.timedate.com** using the ThingsHTTP app that enables communication among devices, websites, and web services without having to implement the protocol on the device level. We specify actions in ThingHTTP[5] app this app works on the HTTP model where the client requests the server for the appropriate information and the server responds back. We had used the url library of the python to make request to server.

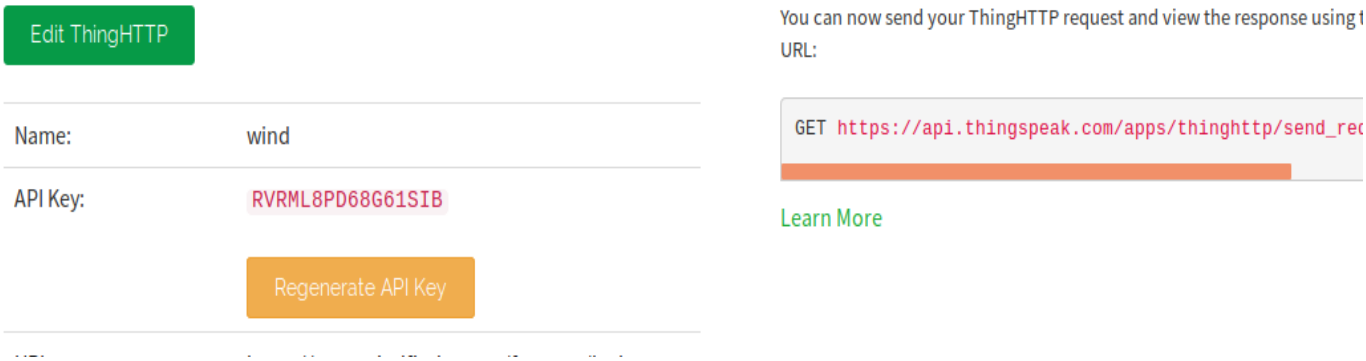


Figure 5.4: ThingsHTTP App Configuration page

5.1.6 Node Number

When recording dataset values are coming from various nodes whuch includes the main (controlling node) and the data logging nodes to distinguish between the data coming from the various nodes and storing them seperately we ned to record the number of node from where data is coming.

- Controlling Node————— 0
- Data Logging Note 1————— 1
- Data Logging Note 2————— 2

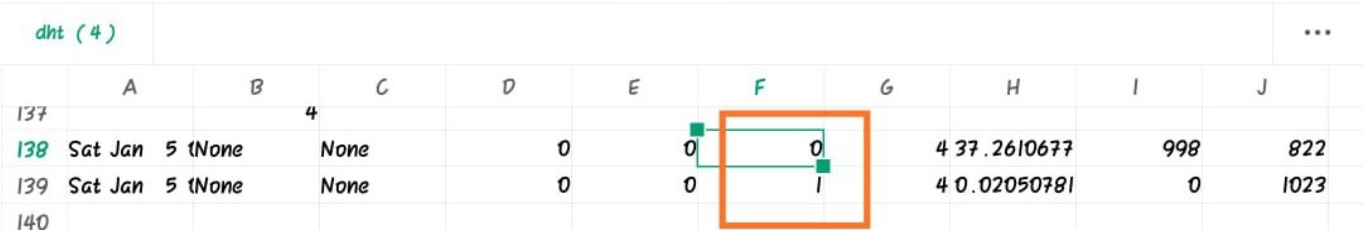


Figure 5.5: Dataset with Node Number

5.2 Image Processing

5.2.1 Capturing Image

To capture the image of the plant we had used the Pi Camera Module that is of the 5 MP and using the Pi camera library of the Python. After capturing the image we had used it to be displayed on the GUI.We had performed image processing techniques to get vatious results.

5.2.2 Processing an Image

In the gardening or Irrigation system it is important to water the plant only when it is green. It is of no use to water the plant if it is dry. If the plant is dry it changes it's colour from the green to the brown or yellow.We had calulated the number of pixels that are green and then used the percentage to find the number of the green content present in the image.Figure shown below is one of the our Output.

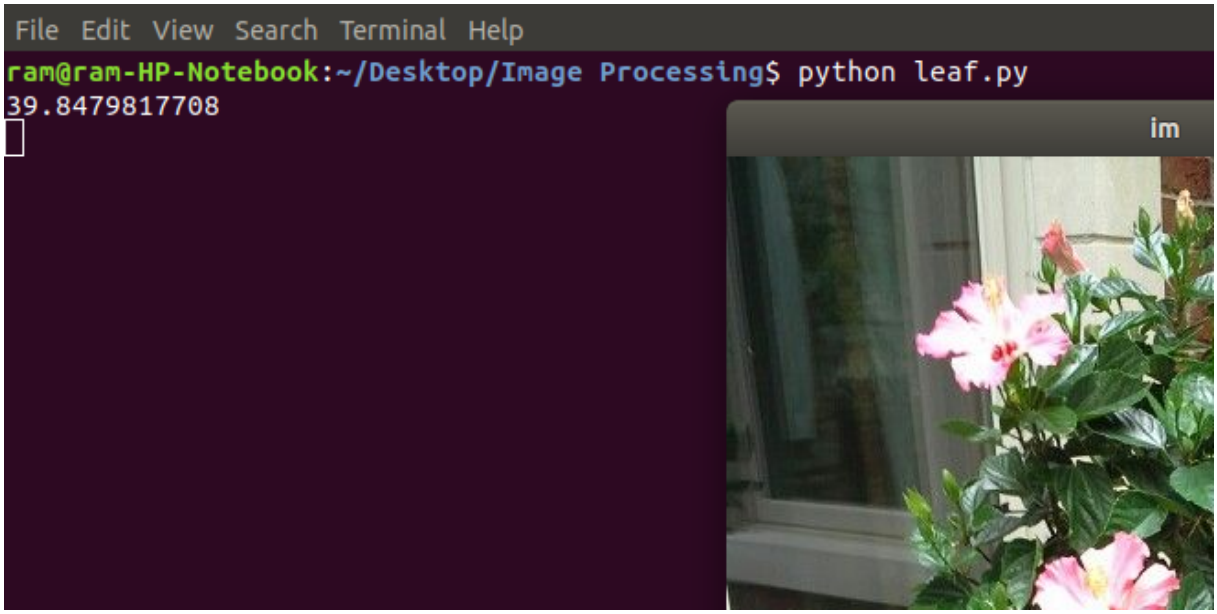


Figure 5.6: Green Colour Content in Image

5.3 Writing data to file

After collecting data from the various sensors. We had wrote all these values to the csv file by opening it in the append mode and using the **file.read()** command of the python we had wrote all the values to the csv file. After the data has been collected successfully and stored in the csv file we had used the various method to accesss the data.

Figure 5.7: Terminal Output when data saved in csv file

5.4 Organizing Node data in file

After collecting data from the both nodes, next problem was to save data in csv file along with the number of node.So that whenever we will have to predict output of relay it becomes easy to send proper command to proper node.
To do so we had to make a sequence on how the data will be transmitted from the data collection nodes to the controlling nodes, synchronization among them plays an important role so as to make sure that data from a particular node get's placed at particular place at the dataset.

5.5 Accessing Data

After recording the dataset and storing it in the proper file format.It was necessary to access the data so that the appropriate preprocessing techniques can be used. To do so we had added options like as view dataset, send dataset to mail ID.To send mail along with dataset as an attachment we had

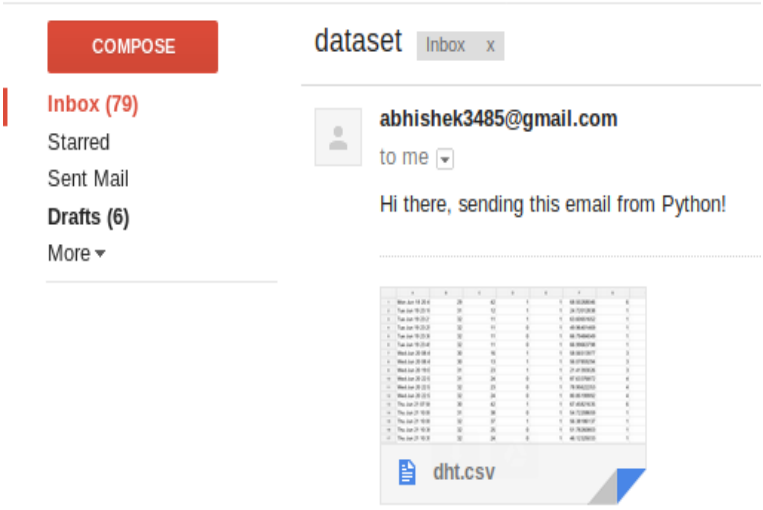


Figure 5.8: Mail sent from the Raspberry Pi

The other way to access the dataset has the HTML webpage on webserver on which the address of the dataset has been included as an hyperlink to the HTML Button. There is option to send the file to the mail-ID for this the python code to send file has been attached to the HTML button.

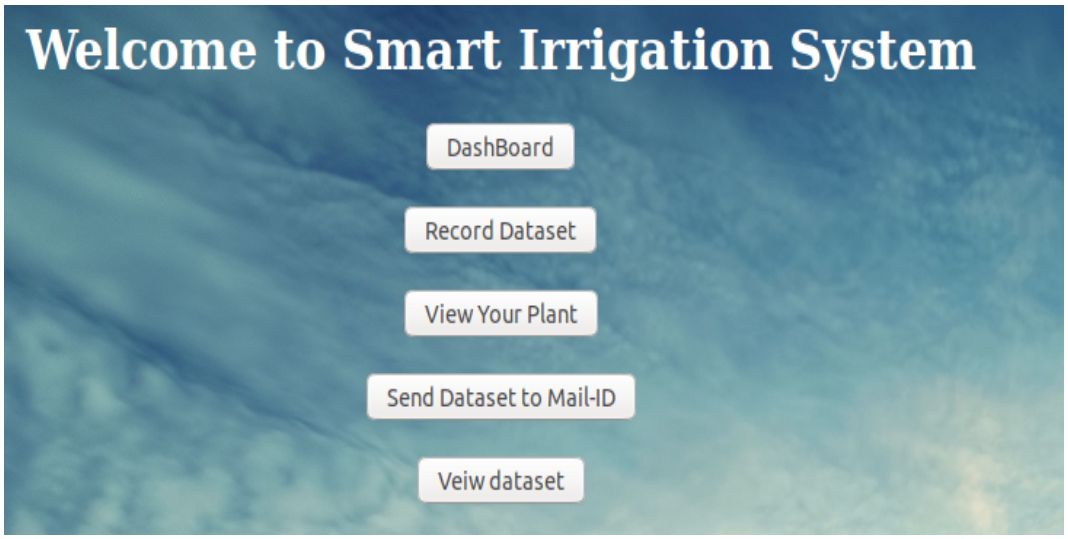


Figure 5.9: Options on the Web Server to Access the dataset

Chapter 6

Data Preprocessing

Since we had colleted the dataset of the sensor values and stored it in a csv file format.Sometimes due to sensor caliberation error the data sent is either unavailable or the data being send it too much high.In our project we had fount that in some of the cases the value of humidity was more than 100% and the percentage of green was too much leass due to less amount of light. To overcome these kind of probelms we had came foreward with the data Preprocessing Techniques.

6.1 Preprocessing our Dataset

6.1.1 Removing the unavailable Values

In our dataset some of the cells were having no values in them.All the machine learning algorithms were unable to process to process the content untill all these were removed. This was caused due to the many of the reasons

- Due to failure of Raspberry Pi in fetching data.
- When an error code get's printed in data set.

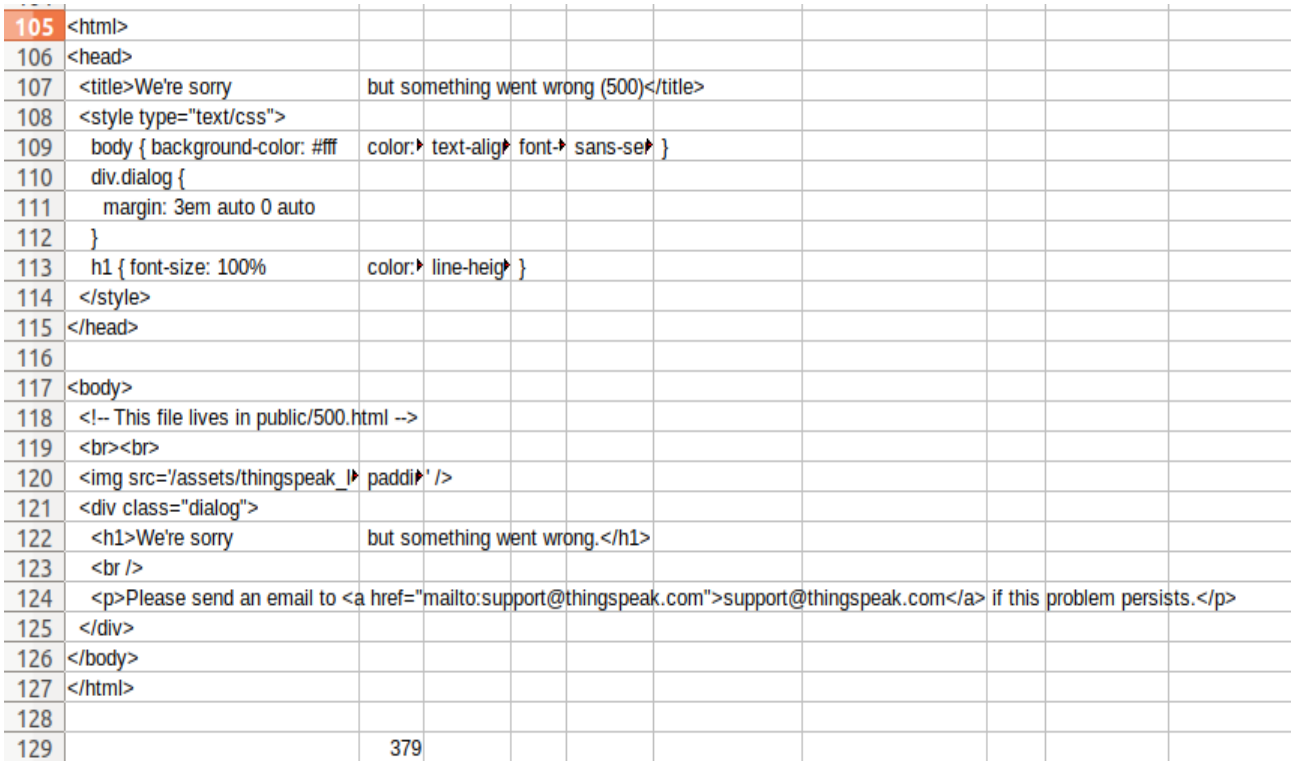
To remove such type of rows we had used the **dataframe.dropna()** command of the pandas library.

65	Sun Jun 24 21:19:59 2018	32	35	0
66	Sun Jun 24 21:20:41 2018	32	35	0
67	Sun Jun 24 21:23:15 2018	None	None	1
68	Mon Jun 25 08:12:51 2018	None	None	1
69	Mon Jun 25 08:14:11 2018	32	27	1

Figure 6.1: None Values in the Dataset

6.1.2 Removing the Redundant Data

Due to some of the problem in running code this message got printed in the dataset to remove the rows having **None** we had used the above approach to remove them.



105	<html>										
106	<head>										
107	<title>We're sorry	but something went wrong (500)</title>									
108	<style type="text/css">										
109	body { background-color: #fff	color:	text-alig	font-	sans-se	}					
110	div.dialog {										
111	margin: 3em auto 0 auto										
112	}										
113	h1 { font-size: 100%	color:	line-heig	}							
114	</style>										
115	</head>										
116											
117	<body>										
118	<!-- This file lives in public/500.html -->										
119	 										
120	<img src='/assets/thingspeak_l	paddi	' />								
121											
122	<h1>We're sorry	but something went wrong.</h1>									
123	 										
124	Please send an email to support@thingspeak.com if this problem persists.										
125											
126	</body>										
127	</html>										
128											
129		379									

Figure 6.2: Error Message in Dataset

6.2 Data Visulisation and Analysis

6.2.1 Soil Moisture level Vs Hours

Soil Moisture means the amount of the water content remaining in the soil, this was the major and most important parameter that helped in deciding wheather to water the plant or not.

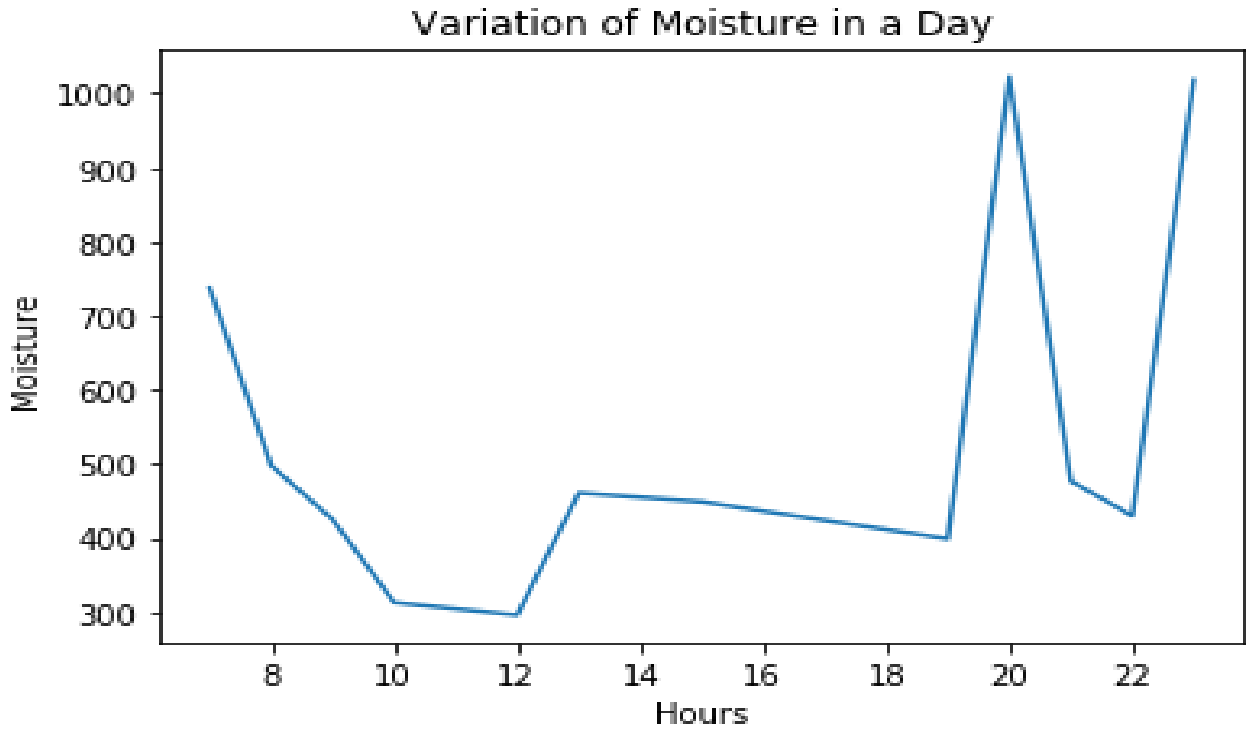


Figure 6.3: Moisture variation with Hours in a Day

Above is the variation of the moisture of soil over a some hours of a day that is taken over a day. Due to the evaporationa and the temperature during a day. The amount of the moisture in a soil decreases.

6.2.2 Temperature variation with vs Date

We are having another parameter that is varying in nature. The variation of temperature over few days have been monitored and corresponding plot had been generated over a certain period of month.

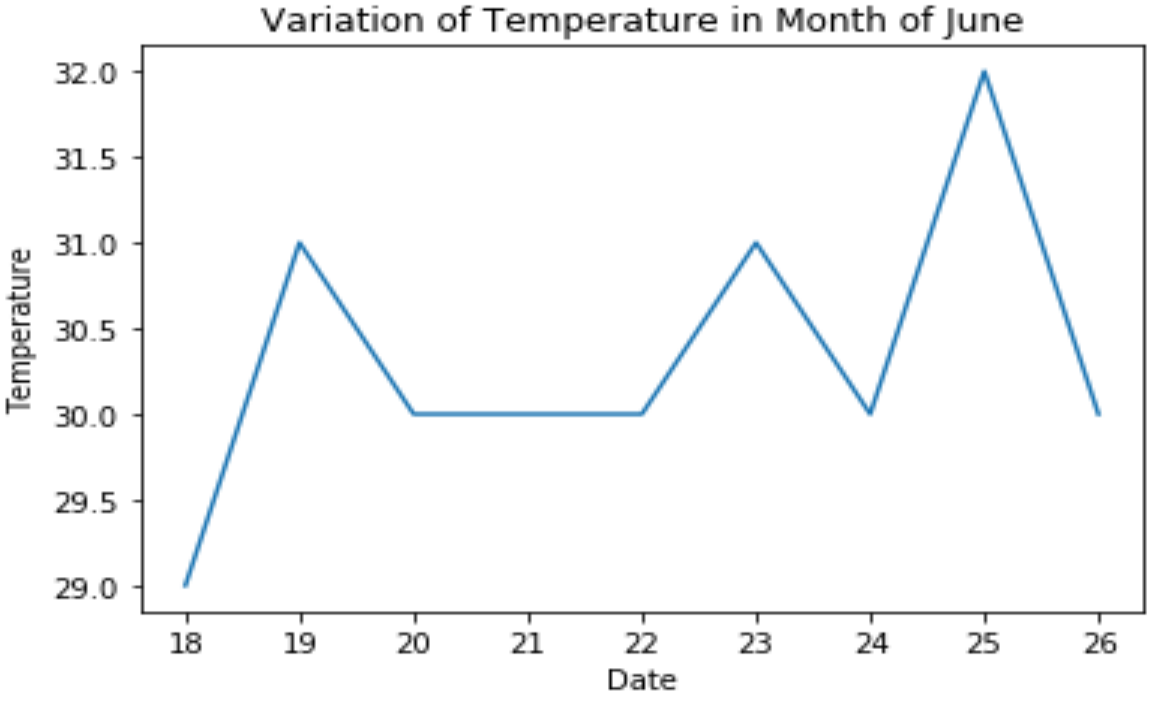


Figure 6.4: Temperature variation over a Month of June

During the month of June there is slight variation in the temperature.Above gaphs shows this nature variation of the temperature over days in month of June.
The variation of the temperature affects the frequency of watering the plant.If the temperature is too much high the amount of water needed is too much high and out plant would require the large amount of water.

6.2.3 Data Set Information

The dataset contains 1235 instances of the data collection from the surrounding of the plant.The whole setup was designed to record the physical conditions in the surrounding of the plant. Data was recorded from June 17 2018 to July 13 2018, and from 22 September 2018 to 5 January 2019. Our Dataset is not labeled (there are no column names available in our dataset. Our dataset has the seven columns out of which the third column is the output of the relay that has the status of the relay (When it is **OFF** or **ON**. The inforaton about the attribtes is as shown below:

- Date (DDMMYYYY) & Time (HH.MM.SS)
- Temperature in degree Celcius

- Air Humidity (in percentage)
- Status of the Relay (to on or off the Water Pump)
- Rain Status (Raining or not Raining)
- Green Percentage (The amount of the Green Colour Available in the captured image)
- Speed of the Wind Blowing the region (fetched from the internet)
- Moisture of the soil
- Water Level Above the soil
- Node Number

Chapter 7

Machine Learning

Machine Learning is a feild of computer science in which we develop algorithms for making the existing systems to take decisions in a more efficient way. Broadly saying, it is divided into two categories: Supervised and Unsupervised learning.

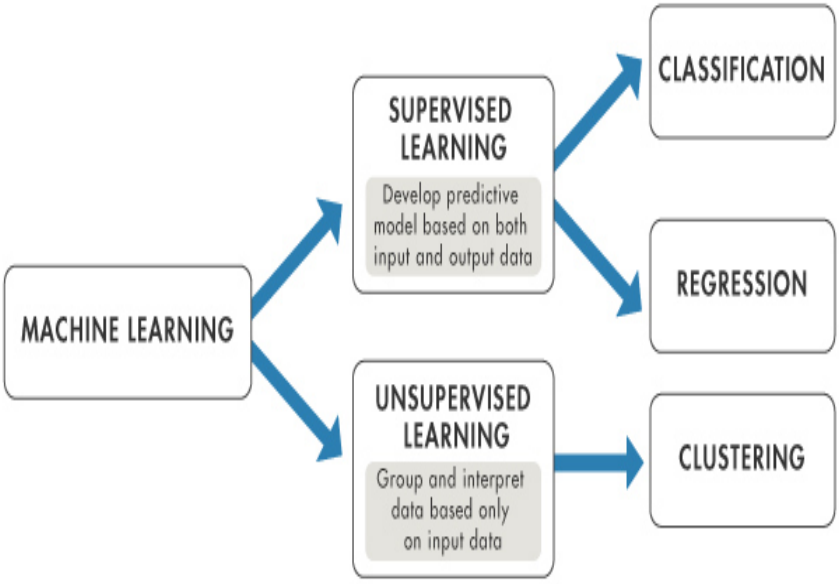


Figure 7.1: Machine Learning overview

In our project, we have utilized the applicaton of Machine Learning in IoT, in which we have collected the data of different sensors in a proper format. Here are some models which we have used in our dataset:

7.1 Logistic Regression

7.1.1 Introduction

Logistic Resgression is a binary classification technique, which is used to predict binary outcomes. (Yes/No, 1/0, True/False). Basically, regression technique involves

minimizing the cost function by determinig the values of weights and biases. Cost is obtained by evaluating error in the predicted value and the truth labels. Gradient Descent is a method which is used in in determinig those values of hyperparameters (weights and baises), at which the error function has global minima. Once the weights are determined, output so predicted is passed into activation functions.

The general form of regression line in fitting the 2D dimensional data is given as

Generally the predicted output is not int the form to directly interpret the results and show it to the user. For this purpose, to implement complex mapping functions, we need activation functions that are non-linear in order to bring in the much needed non-linearity property that enables them to approximate any function. Examples are:

- Sigmoid Function
- ReLU (Rectified Linear Unit)
- Tanh

7.1.2 Implementation in dataset

We have implemented the Logistic Regression using python and Scikit learn library. Numply and pandas were also used to perform all the mathematical and computation operations on the raw dataset. We have removed the column of time and date stamp as it will prevent all the string to float conversion errors.
The test and train sizes are defined using test_train_split function of scikit learn. Test size we have taken is 20% of the total size of dataset, at which the algorithm best performs on our dataset.

7.1.3 Result

Our dataset contains the output labels in the form of 0 and 1, which shows the status of relay. The following outcomes were recorded:

- Accuracy : 68.4%
- Mean Square Error : 0.23469388

7.2 K-Nearest Neighbour

7.2.1 Introduction

k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The k-NN algorithm is among the simplest of all machine learning algorithms. It classifies the test data point into one of the two classes we have, by calculating its distance from each and every existing data points from both the classes. This distance is called "Eucledian Distance". Class is assigned to the data point will contain the nearest neighboure of that data point. The alphabet 'k' denotes the number of neighbours voting on the rest data point.

If K=3, then the labels of three closest classes will be checked and the most common label is assigned.

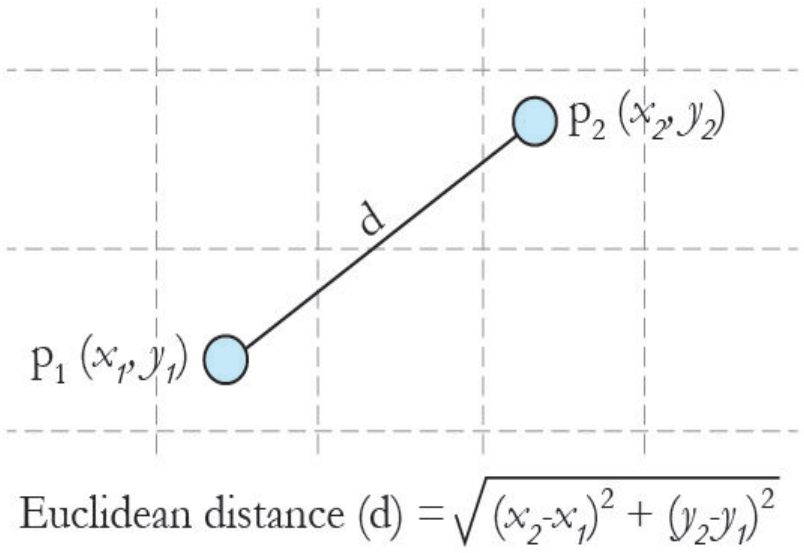


Figure 7.2: KNN algorithm

7.2.2 Result

Since our dataset contains the output labels in the form of 0 and 1, which shows the status of relay, logistic regression prooved to be good. The following outcomes were recorded:

- Accuracy : 87.5%

- f1 Score : 0.90721649

7.3 Support Vector Machines

7.3.1 Introduction

Support Vector Machine is another supervised machine learning algorithm, which also classifies 2 different classes of outputs. It makes the use of supporting planes and separating planes which are collectively known as Hyperplanes.
Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

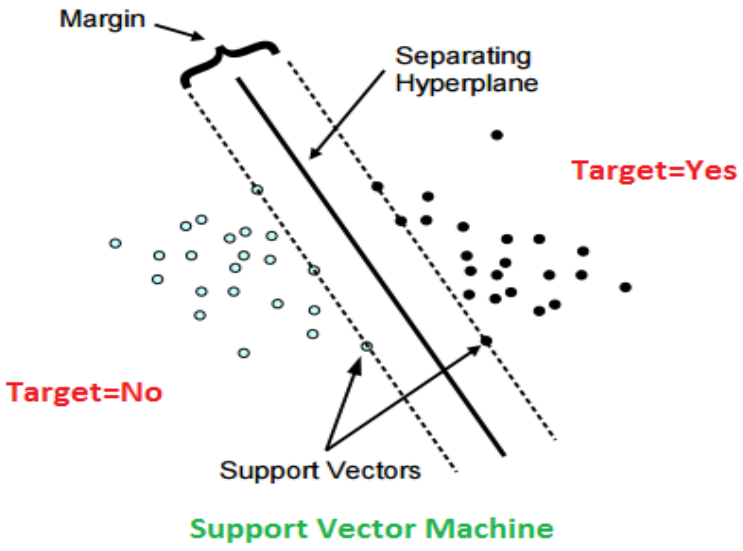


Figure 7.3: SVM planes and Hyperplanes

7.3.2 Result

We have applied the SVM model from scikit-learn machine learning library following outcomes were recorded:

- Accuracy : 0.62364985

- Mean Square Error : 0.23386576

7.4 K-Mean Clustering

k-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.This results in a partitioning of the data space into Voronoi cells. K-means clustering is a classification technique in which 2 different classes are classified by locating their approximated centroids. A line joining their centroids is perpendicularly bisected by a plane which further classsifies the other scattered points into one of the classes. This processes is repeated upto many number of times until unless their centroids stop moving.

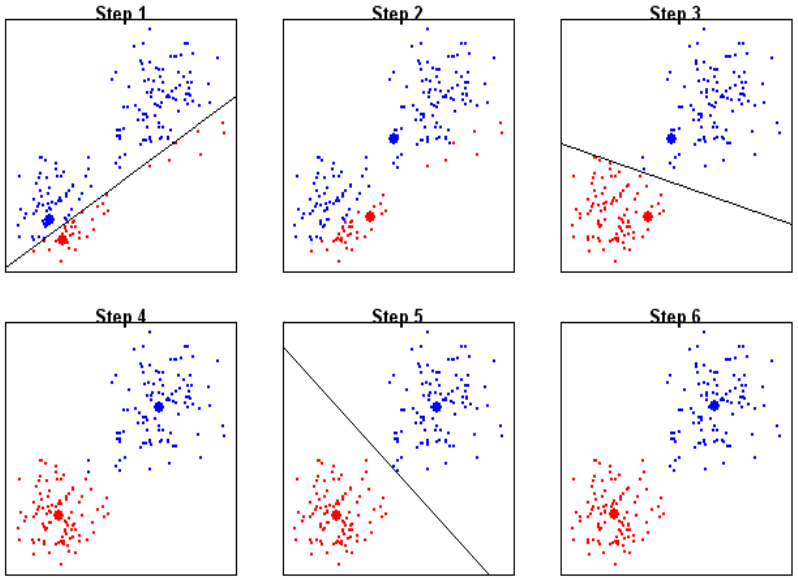


Figure 7.4: K-Means Clustering method

7.4.1 Result

- Mean Square Error : 0.6029
- Score : 0.6029

7.4.2 Selection of best Model

On performing Various operations on our dataset for removing the useless data and after applying various machine learning algorithm, we have selected the K Nearest neighbours algorithm as it best fits our dataset.

The main advantages of KNN algorithm are:

- The main advantage of KNN algorithm is that it performs best on smaller datasets.
- Robust to noisy training data (especially, if we use inverse square of the weighted distance.)
- The cost of learning is zero.
- Performs well on larger datasets

The accuracy we have achieved using KNN is 87.5%

7.5 Conclusion

After applying various models of machine learning on our dataset like SVM, KNN, Logistic Regression and K-means Clustering, we found that KNN algorithm best fitted our data and acheived a test accuracy of 87.5%. We have also used the metrics score to find the score of our KNN algorithm. We have acheived an acore of 0.90721649.

Chapter 8

Aanlysis of ML Models

We had developed various Machine Learning models and had got various results. We had used K-Nearest Neighbor and Support Vector Machine algorithms in our dataset. In this chapter we will sum up the results that we had obtained while using various models.

8.0.1 Choosing Appropriate Model

First problem was to decide which Machine Learning Model we are going to use in our dataset.Since we had our output as the having only two output (either Relay OFF or ON) and there were 13 features on which these were dependent. This was a classifier problem for this purpose we had KNN and SVM algorithm to solve our problem.
In our dataset we had two major parameters that decide when to water plant and when to not. Thinking in broader view we have two features that are important to us and rest are just dependent on them.
Below is a plot that classifies the Water and Moisture among the Rely ON and Relay OFF condition we had used **Pandas,Matplotlib** to make this plot

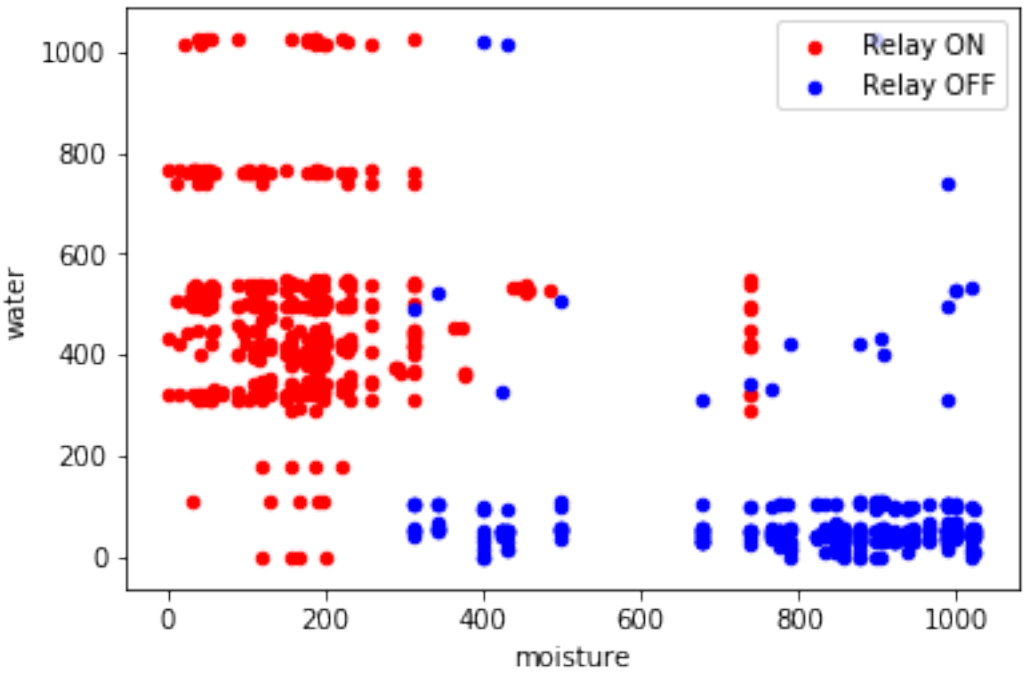


Figure 8.1: Class Classification of Moisture and Water Level

As can be seen clearly this is problem of classifier hence we had used SVM and KNN as our Machine Learning Models. In our project, we have utilized the applicaton of Machine Learning in IoT, in which we have collected the data of different sensors in a proper format. Here are some models which we have used in our dataset:

8.1 K-Nearest Neighbour

8.1.1 Result

We had implemented KNN on our dataset and we had obtained our results as below.

```
ram@ram-HP-Notebook:~/Desktop$ python swmlit_knn.py
F1 Score :- 0.9646243730934285
Accuracy :- 0.9316239316239316
Average_Precision_Score :- 0.9395155500008463
Precision_Score :- 0.9646815550041357
Input Array:- 'temp' 'humidity' 'rain' 'green_per' 'wind' '
Input Array[ 36.  93.   1.  96.  10. 700. 100.]
Predicted Value[1.]
```

Figure 8.2: Result of KNN Algorithm

Below Curve represents the accurate and inaccurate predicted made by the knn algorithm.

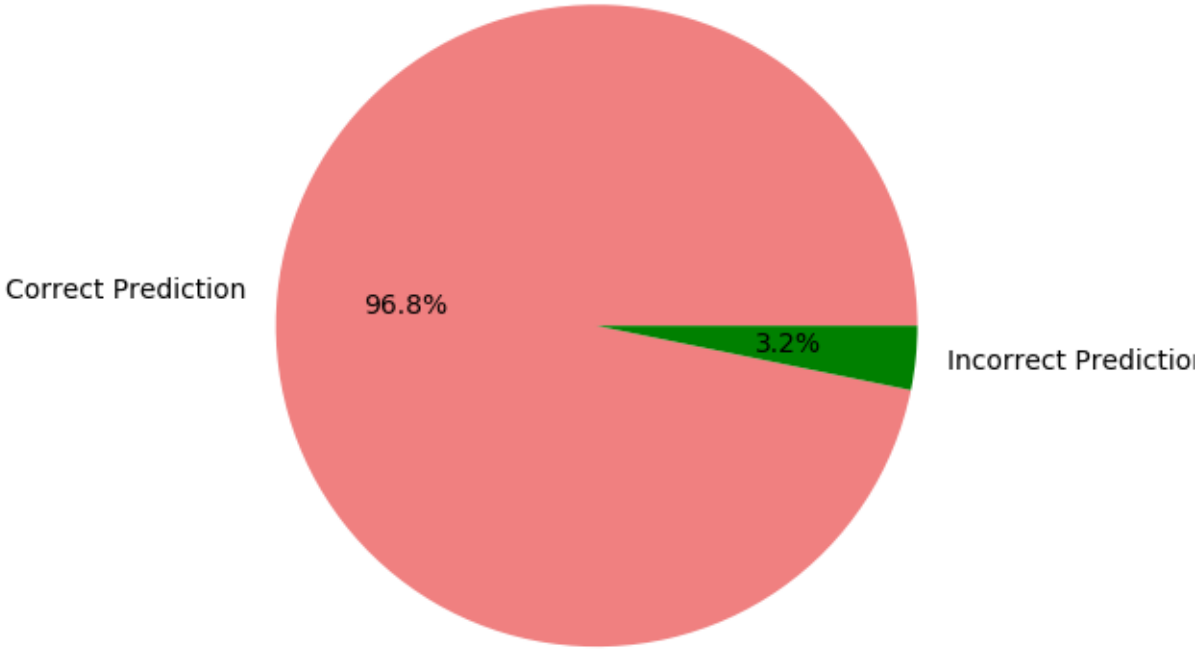


Figure 8.3: Result of KNN Algorithm

8.2 Support Vector Machines

We had also used the Support Vector Machine in our project and had evaluated our SVM model using the various types of kernals.
We had used **RBF,LINEAR,POLYNOMIAL Degree of 2** kernals and made a comprative study.

8.2.1 Results

8.2.2 Comparing Different kernals

Below is the Output of the SVM Model with the LINEAR Kernal.

```
SVC with with Linear Kernal
Test datasize(468,)
Train datasize(311,)
Total Prediction Made311
Correct Prediction:- 283
Incorrect Prediction:- 28
F1 Score :- 0.9098550724637682
Accuracy :- 0.9294871794871795
Average_Precision_Score :- 0.8385760220486909
Precision_Score :- 0.9117549668874172
Input Array:- 'temp' 'humidity' 'rain' 'green_per' 'wind' 'water' 'moisture'
Input Array[ 36.  93.   1.  96.  10. 700. 100.]
Predicted Value[1.]
```

Figure 8.4: Result of SVM with Linear Kernal

Below is the Output of the SVM Model with the RBF Kernal.

```
ram@ram-HP-Notebook:~$ cd Desktop
ram@ram-HP-Notebook:~/Desktop$ python swmlit_svm.py
SVC with RBF kernal
Test datasize(546,)
Train datasize(233,)
Total Prediction Made:- 233
Correct Prediction:- 227
Incorrect Prediction:- 6
F1 Score :- 0.9741914045192734
Accuracy :- 0.9743589743589743
Average_Precision_Score :- 0.9473684210526315
Precision_Score :- 0.9736842105263157
Input Array:- 'temp' 'humidity' 'rain' 'green_per' 'wind' 'water' 'moisture'
Input Array[ 36.  93.   1.  96.  10. 700. 100.]
Predicted Value[1.]
```

Figure 8.5: Result of SVM with Linear Kernal

Below is the Output of the SVM Model with the Poly of Degree 2 Kernal.

```
SVC with Poly Degree 2
Test datasize(546,)
Train datasize(233,)
Total Prediction Made:- 233
Correct Prediction:- 223
Incorrect Prediction:- 10
F1 Score :- 0.9570744288872512
Accuracy :- 0.9120879120879121
Average_Precision_Score :- 0.9186991869918699
Precision_Score :- 0.959349593495935
Input Array:- 'temp' 'humidity' 'rain' 'green_per' 'wind' 'water' 'moisture'
Input Array[ 36.  93.   1.  96.  10. 700. 100.]
Predicted Value[1.]
```

Figure 8.6: Result of SVM with Polynomial Second Degree Kernal

We had ran all these models in our dataset and had found the following accuracies in all these kernals as mentioned below.

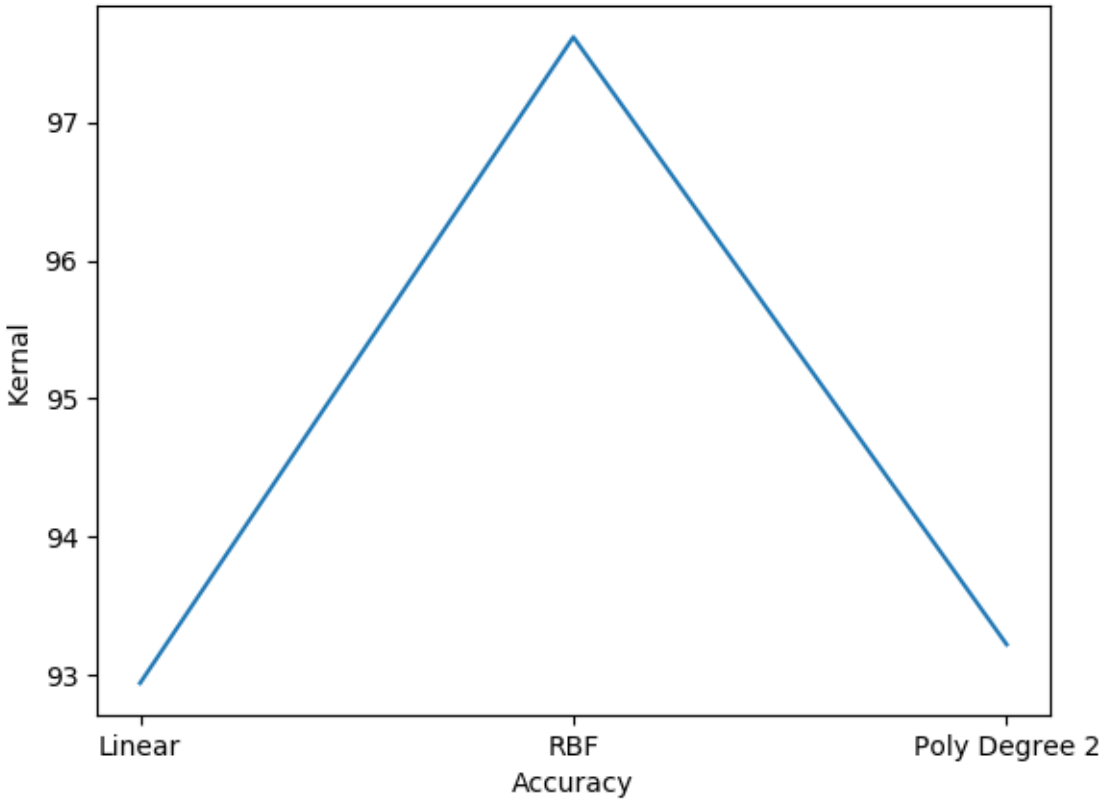


Figure 8.7: Comparison of Accuracy with different kernals

As can be seen that the RBF kernal has the heighest accuracy among all.

8.2.3 Varying the Penalty Parameter C in RBF

RBF stands for Radial Basis Function in machine learning is a popular kernel function used in various kernelized learning algorithms.It is widely used in support vector machine classification. We had varied the Penalty Parameter C which controls the trade off between the achieving a low training error and a low testing error. It helps in generalizing our svm model to unseen data.
Increasing the value of the C should raise the accuracy and this can be seen from the below curve

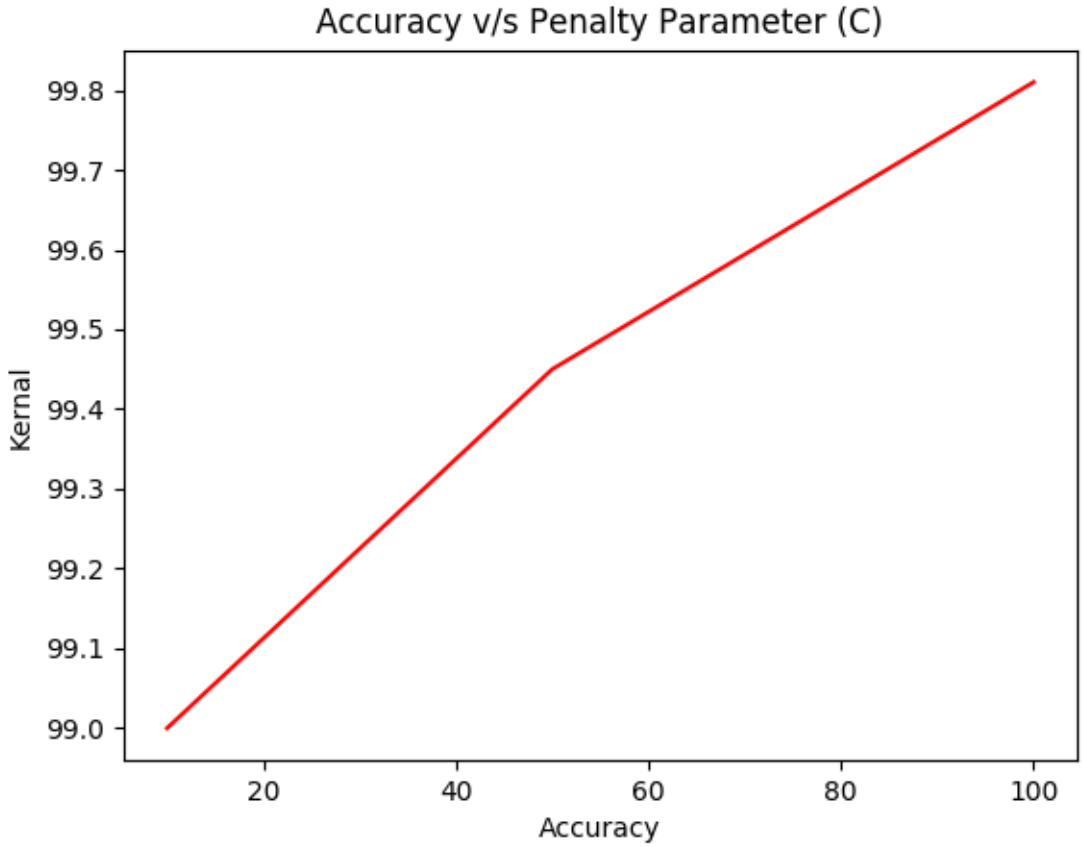


Figure 8.8: Variation of Accuracy with Penalty Parameters

8.3 Conclusion

After applying various models of machine learning on our dataset like SVM, KNN, Logistic Regression and K-means Clustering, we found that KNN algorithm best fitted our data and acheived a test accuracy of 94.5%.We had used the SVM and it's various models using different types of kernals. We have also used the metrics score to find the score of our KNN algorithm. We have acheived an acore of 0.90721649.At the end we had selected the KNN and to go with it as it predicted our Sample input accurately and takes a less amount of time and memory while running which are deciding factors and hence used KNN in our dataset.

Chapter 9

User Interface

Graphical user Interface is an important part of any project as it acts as a bridge between the user and the project. Based upon this concept we too had designed a web Server that acts as a user interface.
Main Features of this interface are-

- Ability to record Dataset from the setup
- Dataset can be sent on the mail of intended user
- Bulit on the IP address of Raspberry Pi-3 it can be accessed from any device on the same network
- User can view the plant image captured by the pi camera
- Dashboard where the Enviroment conditions(viz. Temperature Humidity,Rain,Relay are displayed.The status of relay predicted by the algorithm is also displayed here.

Figure 9.1: Web Page at **192.168.43.175**

9.1 Setting up the Web Server

Setting up the web server on the Raspberry Pi-3 was important in order to set the GUI that can be acessed by devices on the same network.For this purpose we installed the LAMP (Linux Apache and MySQL) on the Pi.By installing the Apache \var\www\html.We created webpage as shown in above image on the IP address of Pi

9.2 Working

The webpage includes the following parts

9.2.1 Displaying Plant Image

Many times user would like to view the image of plant to that he could view the condition of the plant. To provide this we had captured image of the plant and stored at the \var\www\html from there we had displayed it over the web server.

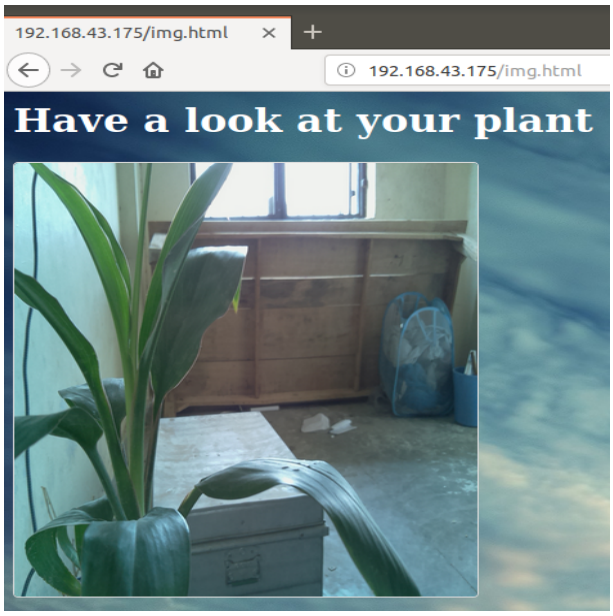


Figure 9.2: Web Page at **192.168.43.175**

9.2.2 Recording the sensor data

This provides user an ability to record the data of the plantś surrounding according to his conveinance.
In this project the python script used to record dataset, is being ran with the help of the php script.
Being server side scripting language php proved to be good laguage in establishing

link between python and the html.When the record dataset button is pressed the php file attched to button on html page activates the python script and thus data set starts to record & dataset is stored by the name dht.csv

9.2.3 Sending the Mail

Since to send mail using python we had already created the python sript. To record data directly form the web we had linked that python script to the html button.// As soon the user presses the html button the control passes to the php script attached to it which in turn triggers the python script to send the dataset on the mail.

9.2.4 Dashboard

It is the most important part of the webserver that provides information like as the Temperature, Humidity,Status of the rain & Predicted status of the relay.
There are six buttons in a table to which there are six python files linked.On clicking these buttons the values sensed by the sensor are displayed on the webpage.

Chapter 10

Conclusion and Future Scope

10.1 Conclusion

We had implemented the project with all the codes and the KNN Clustering algortihm on our project.We had acheived th4e accuracy which varioes from 87.5 to 90.0%. Currently our focus was on a single plant.We had collected the data of the various sensors and manually decided wheather to water plant or not.All this information was saved in a csv file.Based upon this dataset we had created a machine learning model.This model was later upon used to predict the status of the relay thus reducing the manpower.

10.2 Future Scope

This project could be a lot useful for the farmers to predict when to water a crop and when to not water it.This project if could be applied on a large scale and with some improvement could prove to be a lot useful for the farmers with big fields.Certainly today when we are facing a lot of challenges and the lack of manpower in the agriculture sector of the nation technology is only way by which we could give this sector a new boom.Future improvemnts that could be done in the current project are as listed below.

- Applying the project on the large scale with the 4 to 5 nodes acting as a data logging centre
- Improving the GUI for interaction with the users.
- Over the large agricultural fields.In one season of wheat or rice data of the environental conditions and instances when we have to water the field based upon this large dataset we can design a system to predict wheather to water field or not.

Chapter 11

Running Codes

11.1 How to run the code

Running and diagnosing the complete setup is an important task in order for its efficient deployment in agricultural system. The user friendly GUI contains simple buttons which runs all the backend tasks hidden from the user. Although the complete procedure to control the system is described below.

- switch.py - This python script cosists of the code that is used to record the data from the various sensors and storing them to the dataset. To run this code this needs to transferred to the raspberry pi using the file transfer proctocol using FileZilla and then running it in the terminal of the raspberry pi by typing sudo python switch.py
- knn_Rpi.py - This python script cosists the machine learning model that could be run on the Raspberry Pi. KNN clustering model has been used and it's code is written in this python script.This script takes the values from the sensors and based upon them it predicts the status of the relay. To run this code this needs to transferred to the raspberry pi using the file transfer proctocol using FileZilla and then running it in the terminal of the raspberry pi by typing sudo python knn_Rpi.py
- send_mail.py and send_image_mail.py - These Python Scripts are useful and are used to send the dataset and the image of the plant as an mail to the designated mail id.

  These should be ran on the raspberry pi by following the above method. All the above files need to be in the same directory on the Raspberry Pi.

- index.html, run_send_mail.php run_main_script.php,Display Image.phpl- These files are files for the backend of the webpage that is running on the Raspberry Pi. These files need to bbe stored in the folder at location \var\html\www

References

[1] *Important preprocessing methods*
http://www.uta.fi/sis/tie/tl/indexDatamining4.pdf

[2] *DHT Sensor*
https://howtomechatronics.com/tutorials/arduino/
dht11-dht22-sensors-temperature-and-humidity-tutorial-using-arduino/

[3] *Raspberry Pi 3 Camera Module*
https://uk.pi-supply.com/products/raspberry-pi-camera-board-v1-3-5mp-

[4] *Raspberry Pi 3 Camera Module*
https://store.arduino.cc/usa/arduino-uno-rev3

[5] *ThingsHTTP*
https://www.mathworks.com/help/thingspeak/thinghttp-app.html

[6] *Danilo Bzdok, Martin Krzywinski, Naomi Altman. Machine learning: Supervised methods, SVM and kNN. Nature Methods, Nature Publishing Group, 2018, pp.1*
https://hal.archives-ouvertes.fr/hal-01657491/document

[7] *Comparing Accuracy of K-Nearest-Neighbor and Support-Vector-Machines for Age Estimation*
http://www.ijettjournal.org/2016/volume-38/number-6/
IJETT-V38P260.pdf

[8] *Comparison: KNN SVM Algorithm*
https://www.ijraset.com/fileserve.php?FID=11852

[9] *Support-vector machine*
https://en.wikipedia.org/wiki/Support-vector_machine

[10] *SVM*
http://apps.nrbook.com/empanel/index.html#pg=883

[11] *Choosing the right estimator*
https://scikit-learn.org/stable/tutorial/machine_learning_map/