AY 2023-24

V Semester

ECL502: PROJECT PHASE I

A PROJECT REPORT
ON

ARYA.AI (Agriculture Resource Yield Analytics)

Bachelor of Technology

in

Electronics and Communication Engineering

By

DIVYANSH SRIVASTAVA (21216)
RISHAV MISHRA (21235)
SHREYASH JADHAO (21241)

SCHOOL OF ELECTRONICS

**INDIAN INSTITUTE OF INFORMATION TECHNOLOGY UNA
HIMACHAL PRADESH**

NOVEMBER 2023

BONAFIDE CERTIFICATE

This is to certify that the project titled **ARYA.AI (Agriculture Resource Yield Analytics)** is a bonafide record of the work done by

DIVYANSH SRIVASTAVA (21216)
RISHAV MISHRA (21235)
SHREYASH JADHAO (21241)

in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in ELECTRONICS AND COMMUNICATION ENGINEERING of the INDIAN INSTITUTE OF INFORMATION TECHNOLOGY UNA, HIMACHAL PRADESH, during the year 2023 - 2024.

under the guidance of

DR. ANKUR KUMAR

School Of Electronics

Indian Institute Of Information Technology Una

Project viva-voce held on: _________________

Internal Examiner   External Examiner

ORIGNALITY / NO PLAGARISM DECLARATION

We certify that this project report is our original report and no part of it is copied from any published reports, papers, books, articles, etc. We certify that all the contents in this report are based on our personal findings and research and we have cited all the relevant sources which have been required in the preparation of this project report, whether they be books, articles, reports, lecture notes, and any other kind of document. We also certify that this report has not previously been submitted partially or as whole for the award of degree in any other university in India and/or abroad.

We hereby declare that, we are fully aware of what constitutes plagiarism and understand that if it is found at a later stage to contain any instance of plagiarism, our degrees may be cancelled.

DIVYANSH SRIVASTAVA (21216)
RISHAV MISHRA (21235)
SHREYASH JADHAO (21241)

ABSTRACT

In response to the multifaceted challenges confronting modern agriculture, ARYA.AI (Agriculture Resource Yield Analytics) emerges as a revolutionary solution that seamlessly integrates conventional farming wisdom with cutting-edge technology. This project addresses the shortcomings of traditional farming practices by leveraging the power of technology, data analytics, and automation to provide farmers with data-driven insights and recommendations.

ARYA.AI employs a comprehensive approach, combining various sensors, Internet of Things (IoT), machine learning (ML), and a user-friendly mobile app. The project collects an array of sensor data, encompassing temperature, humidity, precipitation, wind, and nutrient levels on a farm. Machine learning algorithms analyze this data to deliver individualized crop suggestions, maximize resource usage, and identify anomalies or illnesses in crops. The integration of an autonomous irrigation system streamlines operations, and a sensor anomaly detection feature ensures the system's reliability.

The project report delves into the technical intricacies of ARYA.AI, detailing the deployment of the prototype farm and exploring the transformative potential of this innovative solution in the realm of agriculture. By addressing the urgent need to revolutionize farming practices, ARYA.AI aims to empower farmers with real-time insights, precise recommendations, and proactive disease detection, ultimately leading to informed decision-making, resource optimization, and improved crop yields and profitability.

Moreover, ARYA.AI recognizes the importance of direct farmer-consumer interactions, fostering fairer transactions and reducing reliance on intermediaries. In a world increasingly prioritizing sustainable agriculture, this project endeavors to usher in a new era of smart, data-driven farming that not only benefits farmers and consumers but also contributes to environmental sustainability.

Key achievements of the project include the implementation of machine learning models for sensor anomaly detection, crop prediction, and fertilizer prediction. These models are seamlessly integrated into an Android app featuring an intuitive user interface (UI), enhancing accessibility for farmers. By combining advanced technologies and traditional farming knowledge, ARYA.AI stands as a beacon of innovation in the agricultural landscape, promising a brighter, more sustainable future for farming communities and the environment alike.

Keywords: Precision farming, sensors, irrigation ,moisture,IOT,ML,Android

ACKNOWLEDGEMENT

I would like to thank the following Estate maintenance office, Mr. Ayush(civil), Mr. Nitin Dhillon(electrical) for their support and guidance without whom the completion of this project in fruition would not be possible.

I would like to express my sincere gratitude and heartfelt thanks to Dr. Ankur Kumar for his unflinching support and guidance, valuable suggestions and expert advice. His words of wisdom and expertise in subject matter were of immense help throughout the duration of this project.

I also take the opportunity to thank our Director and all the faculty of School of Computing/Electronics, IIIT Una for helping me by providing necessary knowledge base and resources.

I would also like to thank my parents and friends for their constant support.

DIVYANSH SRIVASTAVA (21216)
RISHAV MISHRA (21235)
SHREYASH JADHAO (21241)

TABLE OF CONTENTS

LIST OF ACRONYMS

ARDUINO	Popular open-source electronics platform
IoT	Internet of Things
ThinkSpeak	data streams in the cloud
ML	Machine Learning
XML	Extensible Markup Language
SVM	Support Vector Machine
KNN	K Nearest Neighbour
UI	User Interface

LIST OF FIGURES

Chapter 1
Introduction

1.1. Brief Introduction Of the Project

ARYA.AI, or Agriculture Resource Yield Analytics, is a groundbreaking project that merges traditional farming knowledge with advanced technology. Designed to address the challenges of modern agriculture, it integrates sensors, IoT, machine learning, and a user-friendly app. ARYA.AI aims to empower farmers with data-driven insights, proactive disease detection, and optimized resource usage, ushering in a new era of smart, sustainable farming.

1.2. Tech Stack Used

1. **Hardware:**
    - NodeMCU ESP8266 Board
    - Capacitive Soil Moisture Sensor
    - DS18B20 Waterproof Temperature Sensor
    - DHT22 Temperature and Humidity Sensor
    - PIR Motion Sensor
    - Rain Detector Sensor
    - 0.96-Inch OLED Display
    - Volt Single-Channel Relay Module
    - Volt Buzzer
    - Volt DC Pump Motor
2. **Machine Learning:**
    - Autoencoders for Sensor Anomaly Detection
    - Linear Interpolation
    - KNN Imputer
    - Deep Learning Neural Network for Crop Prediction
    - Neural Network for Fertilizer Prediction
    - Comparison of Machine Learning Models (Decision Trees, SVM, Random Forest)
3. **Mobile Application:**
    - Kotlin Programming Language
    - Android Studio for App Development
    - Firebase for Backend Database
    - Firebase Authentication for User Authentication
    - 4. Internet of Things (IoT):
    - Blynk IoT Platform/ThingSpeak for Remote Monitoring and Control

- ESP8266 NodeMCU for IoT Connectivity

5. Data Processing and Analysis:

- Python for Data Processing
- Pandas for Data Manipulation
- Scikit-learn for Machine Learning Models
- TensorFlow/Keras for Deep Learning

6. Version Control:

- Git for Version Control

7. Cloud Services:

- Firebase for Backend Services and Authentication

8. Integration:

- Integration of Machine Learning Models with IoT Devices
- Integration of Firebase with Android App

9. Development Environment:

- Jupyter Notebooks for Machine Learning Development
- Android Studio for Mobile App Development
- Arduino IDE for IoT Device Programming

10. Communication:

- HTTP/HTTPS for IoT Device Communication
- Firebase Cloud Messaging (FCM) for App Notifications

11. Database:

- Firebase Realtime Database for Data Storage

12. User Interface (UI):

- Android XML for App UI Design
- Blynk App for IoT Device Interface

Chapter 2
Problem Statement, Motivation and Objective

2.1 Problem statement

Traditional farming practices lack precision, leading to inefficient resource utilization and reduced crop yields. Inadequate irrigation systems, imprecise crop predictions, and suboptimal fertilizer application contribute to increased water wastage and economic losses for farmers. Recognizing these challenges, our project seeks to revolutionize agriculture by integrating IoT and machine learning. The aim is to empower farmers with real-time insights, automated irrigation control, and data-driven decision-making, ultimately fostering sustainable and efficient farming practices.

2.2 Motivation for the Project

The motivation behind this project stems from the urgent need to revolutionize conventional farming methods. By integrating advanced technologies such as IoT and machine learning, we aim to empower farmers with real-time insights, precise recommendations, and automated control over irrigation processes.

Figure 2.1: Wastage of Crops

2.3 Proposed solution

2.3.1 Preface

Our solution combines IoT and machine learning to create a smart agriculture system, enabling precise resource management. Through real-time sensor data, automated irrigation, and accurate predictions, we strive to enhance crop yields, reduce water wastage, and usher in a new era of sustainable and efficient farming practices.

2.3.2 Brief introduction of Work Done

- Implemented NodeMCU ESP8266 for IoT connectivity.
- Integrated capacitive soil moisture, temperature, humidity, motion, and rain detectors for comprehensive monitoring.
- Utilized a 0.96-inch OLED display for real-time data visualization.
- Applied Autoencoders for anomaly detection and filled missing values in temperature sensor data.
- Employed a deep learning neural network for accurate crop prediction.
- Developed a separate neural network model for fertilizer prediction.
- Implemented the Android app in Kotlin for user-friendly interfacing.
- Utilized Firebase as the backend database for secure data storage.
- Integrated Blynk IoT Platform for remote monitoring and control.
- Established a 5V relay module for controlling the water pump.
- Enabled automatic irrigation initiation based on soil moisture levels.
- Implemented an alert system with a 5V buzzer for anomaly detection.
- Evaluated and compared machine learning models, including decision trees, SVM, and Random Forest.
- Selected and implemented the most accurate model for crop prediction.
- Used Python with Pandas for data processing and manipulation.
- Employed TensorFlow/Keras for developing and training machine learning models.
- Incorporated Firebase Authentication for secure user access.
- Ensured seamless communication between IoT devices and the Android app.
- Developed Android XML for an intuitive and user-friendly interface.
- Facilitated remote monitoring, anomaly alerts, and irrigation control through the app.
- Utilized a variety of sensors, including capacitive soil moisture, DS18B20 temperature, DHT22 humidity and temperature, PIR motion, rain detector, and 0.96-inch OLED display.
- Integrated a 5V DC pump motor for automated irrigation.

This comprehensive set of technical functionalities represents the seamless integration of IoT, machine learning, and mobile application development, providing farmers with advanced tools for precise agricultural management.

2.4 Objectives

The objective of this project is to develop an application which has been made accomplishing the following objectives:

- Integrate diverse sensors for comprehensive data collection.
- Develop machine learning models for crop recommendations.
- Implement an automatic irrigation system for resource optimization.
- Utilize drone technology for proactive crop monitoring.
- Ensure seamless IoT integration for real-time data communication.
- Create a user-friendly mobile app for farmers.
- Implement sensor anomaly detection for data reliability.
- Establish a scalable cloud backend for data storage and management.
- Develop a marketplace within the app for direct farmer-consumer transactions.
- Enhance agricultural productivity through data-driven insights.

Chapter 3
Review Of Literature and Preliminary Study

3.1. Hardware Part

3.1.1 FIXED TIME IRRIGATION CONTROL SYSTEM

The use of IoT in agriculture has gained significant attention in recent years due to its potential to improve crop yields and reduce water usage. Several studies have been conducted on the development and implementation of IoT-based smart agriculture systems.

In a study conducted by Kumar and Senthil Kumar (2019), an IoT-based automated irrigation system was developed for paddy crops. The system used sensors to monitor soil moisture and temperature, and a microcontroller was used to control the irrigation system. The study reported that the system reduced water usage by 50% and increased crop yield by 20%.

3.1.2 DEVOLEPMENT OF A LOW-COST COMMUNITY BASED REAL TIME REDUCTION OF WATER WASTAGE

Another study by Rathore et al. (2020) developed an IoT-based smart irrigation system for tomato crops. The system used various sensors to monitor soil moisture, temperature, and humidity. The data collected by the sensors was transmitted to a central control unit that determined the optimal irrigation schedule. The study reported that the system reduced water usage by 60% and increased crop yield by 25%.

A study conducted by Ahlawat et al. (2021) developed an IoT-based smart agriculture system that used image processing to detect and classify different weeds in a wheat field. The system used a camera to capture images of the field, and the images were processed using machine learning algorithms. The system could then identify and classify the different weeds and control them using precision spraying. The study reported that the system reduced herbicide usage by 70% and increased crop yield by 15%.

3.1.3 LESSER USE OF HERBICIDES AND PESTICIDES

The literature review indicates that IoT-based smart agriculture systems have the potential to improve crop yields and reduce water usage. The use of sensors, microcontrollers, and other IoT-enabled devices can provide real-time monitoring and control of the irrigation system, enabling farmers to make informed decisions about their crops. The use of machine learning and image processing can also provide precision agriculture solutions, which can reduce the use of herbicides and pesticides and improve crop yields.

3.2 Software Part

3.2.1 Python Programming

3.2.1.1 Introduction

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming.

3.2.1.2 Libraries and Frameworks

- **Numpy:** NumPy offers comprehensive mathematical functions, random number generators, linear algebra routines, Fourier transforms, and more. Interoperable.
- **Pandas:** Pandas is a Python package that provides fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data.
- **Pickle:** The pickle module implements binary protocols for serializing and de-serializing a Python object structure. “Pickling” is the process whereby a Python object hierarchy is converted into a byte stream, and “unpickling” is the inverse operation, whereby a byte stream (from a binary file or bytes-like object) is converted back into an object hierarchy.
- **Sklearn:** Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python.
- **Keras:** Keras is the high-level API of TensorFlow and an approachable, highly-productive interface for solving machine learning problems, with a focus on modern deep learning. It provides essential abstractions and building blocks for developing and shipping machine learning solutions with high iteration velocity.
- **Tensorflow:** TensorFlow is an open-sourced end-to-end platform, a library for multiple machine learning tasks.

3.2.2 Deep Learning

3.2.2.1 Introduction

Deep learning is **a method in artificial intelligence (AI) that teaches computers to process data in a way that is inspired by the human brain**. Deep learning models can recognize complex patterns in pictures, text, sounds, and other data to produce accurate insights and predictions. It is an important element of data science, which includes statistics and predictive modeling. Deep Learning is the driving force behind the notion of self-driving automobiles that are autonomous. Deep Learning technologies are actually "learning machines" that learn how to act and respond using millions of data sets and training.

3.2.2.2 Algorithms

- **Artificial Neural Network:** An artificial neural network (ANN) is a type of machine learning model inspired by the structure and function of the human brain. It is a computational model composed of interconnected nodes, called artificial neurons, which work together to process and learn from input data.
- **Convolutional Neural Network:** CNNs have an architecture that includes convolutional layers, pooling layers, and fully connected layers. The convolutional layers perform a convolution operation, where a set of learnable filters are applied to the input image to extract features.

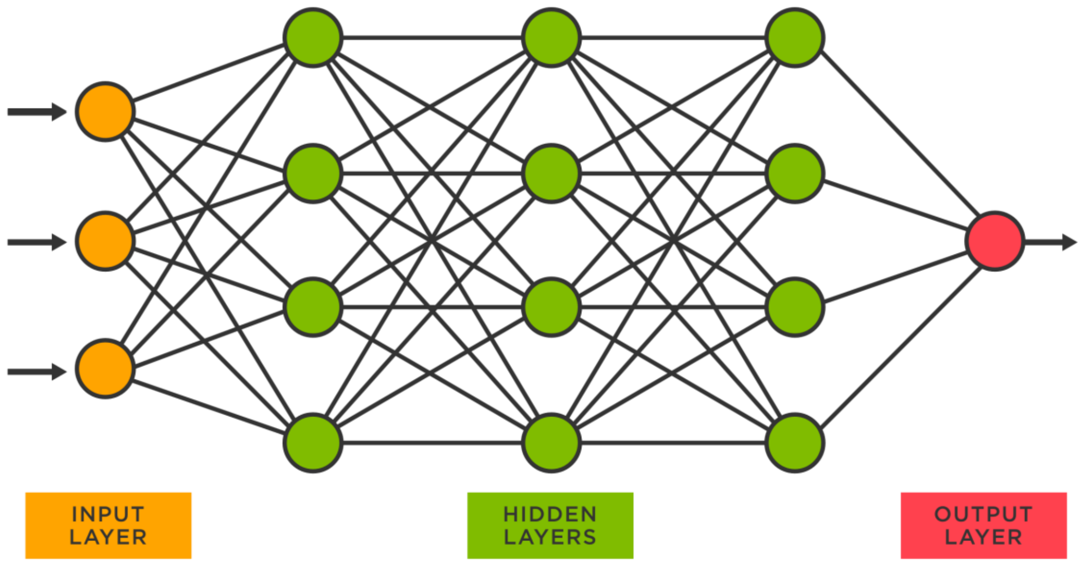


Figure 3.1: Neural Network Diagram

3.2.3 Autoencoders:

Autoencoders are unsupervised neural network models designed for data compression and feature learning. Consisting of an encoder and a decoder, they aim to reproduce the input data at the output, with an intermediate layer representing a compressed form of the input. In our project, Autoencoders play a pivotal role in sensor anomaly detection. By training on a dataset derived from 90 days of temperature sensor readings, Autoencoders learn the underlying patterns and aid in filling missing values.

3.2.4 Machine Learning Model:

The machine learning model deployed in the project focuses on accurate crop prediction. After comparing various models like decision trees, SVM, and Random Forest, a deep learning neural network was chosen for its superior performance. This neural network processes historical data to predict the crops suitable for the farm under given conditions, contributing to informed decision-making.

3. Winsorization for Outlier Removal:

Winsorization is a statistical method used to address outliers in the dataset. In our project, it plays a crucial role in enhancing the accuracy and reliability of the machine learning models. By replacing extreme values with values closer to the mean, winsorization ensures that outliers do not unduly influence the model training, contributing to a more robust and accurate system.

4. Android Programming with Kotlin:

Kotlin, a modern programming language for Android development, was employed to create a user-friendly and efficient mobile application. It offers concise syntax, enhanced readability, and interoperability with Java. Our Android app, developed in Kotlin, provides seamless interfacing for users to remotely monitor and control the smart agriculture system.

5. Firebase Backend:

Firebase, a comprehensive mobile and web development platform, was utilized for backend services, particularly authentication and Database Management anlong with Mongo DB. Firebase Authentication ensures secure user access to the Android app, safeguarding sensitive data. By providing a scalable and reliable authentication mechanism, Firebase contributes to the overall security and integrity of the smart agriculture system.

Chapter 4
METHODOLOGY

HARDWARE:

The methods for a IoT based smart agriculture monitoring and automatic irrigation system typically involves several steps, including designing and building the hardware and software components of the system, testing and evaluating the system in different scenarios, and implementing the system in a real-world environment.

1) System Architecture:

The proposed IoT-based smart agriculture and automatic irrigation system consists of various sensors, actuators, and other IoT-enabled devices that are integrated into the irrigation system. The system is controlled by a central control unit, which receives data from the sensors and determines the optimal irrigation schedule based on the environmental conditions.

2) Sensors and Actuators:

The sensors used in the system include soil moisture sensors, temperature sensors, humidity sensors, and rainfall sensors. The soil moisture sensor measures the moisture content of the soil, while the temperature sensor measures the temperature of the soil. The humidity sensor measures the humidity of the air, and the rainfall sensor measures the amount of rainfall. The actuators used in the system include solenoid valves, which control the flow of water to the irrigation system.

3) Microcontroller:

The microcontroller used in the system is an Arduino board, which receives data from the sensors and controls the solenoid valves. The microcontroller is programmed to adjust the irrigation schedule based on the data received from the sensors.

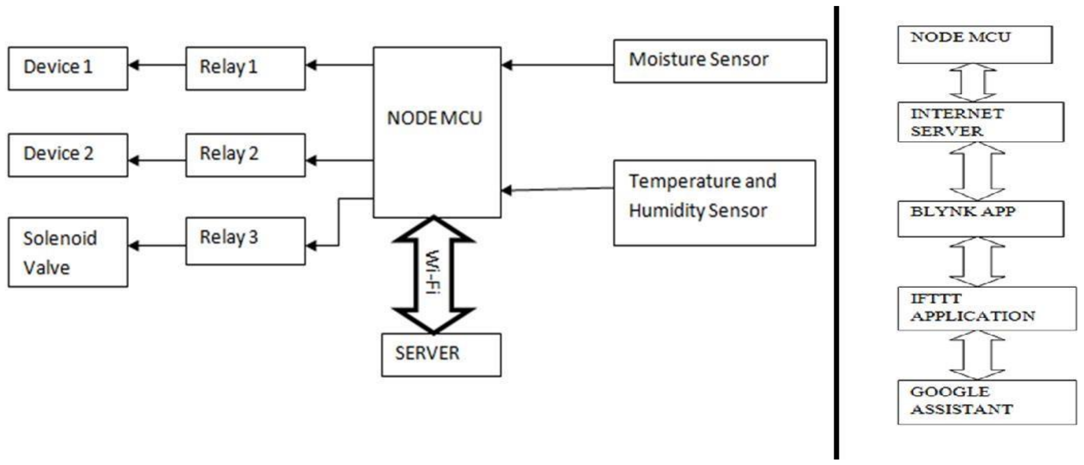


Fig 4.1: Block diagram of the Connections on board

4) Cloud Connectivity:

The system is connected to the cloud, which allows farmers to remotely monitor the system using a smartphone or a computer. The cloud also stores data collected by the sensors, which can be used for analysis and decision-making.

5) Installation:

The sensors and actuators are installed in the field, and the microcontroller is placed in a weatherproof enclosure. The solenoid valves are connected to the irrigation system, and the microcontroller is connected to the sensors and the solenoid valves. The system is powered by a solar panel, which provides renewable energy to the system.

6) Data Collection and Analysis:

Data collected by the sensors is transmitted to the cloud, where it is stored and analyzed. The data is analyzed using machine learning algorithms, which can provide insights into the crop yields and the effectiveness of the irrigation system.

Overall, the methods section outlines the different components of the IoT-based smart agriculture and automatic irrigation system, how they are installed and connected, and how data is collected and analyzed. The proposed system uses sensors, microcontrollers, and cloud connectivity to provide real-time monitoring and control of the irrigation system, enabling farmers to make informed decisions about their crops.

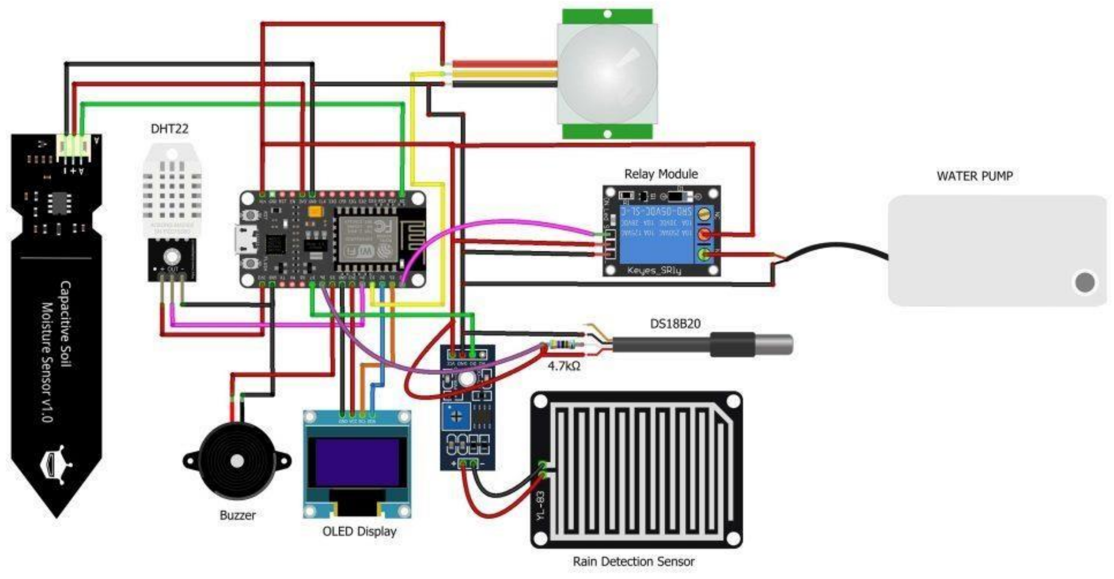


Fig 4.2: Basic Model of the System(software generated)

SOFTWARE:

- **Autoencoders for Anomaly Detection:**
  - Preprocessed sensor data using Python and Pandas.
  - Applied Autoencoders to learn patterns and fill missing values in temperature sensor readings.
  - Achieved enhanced accuracy in anomaly detection by training the model on a 90-day dataset.
- **Neural Network for Crop Prediction:**
  - Conducted a comparative analysis of machine learning models, including decision trees, SVM, and Random Forest.
  - Selected a deep learning neural network for its superior accuracy.
  - Trained the neural network on historical data to predict crops based on environmental conditions.
- **Fertilizer Prediction Model:**
  - Developed a separate neural network model for fertilizer prediction.
  - Trained the model on relevant environmental and soil data.
  - Integrated the model to provide data-driven fertilizer recommendations for optimal crop growth.
- **Integration of Hardware and Software**
- **Data Processing and Analysis:**
  - Used Python for data processing and analysis.
  - Employed TensorFlow/Keras for developing and training machine learning models.
  - Android Application Development:
  - Developed the Android app in Kotlin for user-friendly interfacing.
  - Ensured seamless integration with Firebase for real-time data updates.
  - Utilized Firebase Authentication for secure user access.
- **Communication between Databases:**
  - Established a connection between Firebase and MongoDB for efficient data storage.
  - Enabled data synchronization to maintain consistency between the two databases.
  - Facilitated comprehensive data retrieval and storage capabilities.

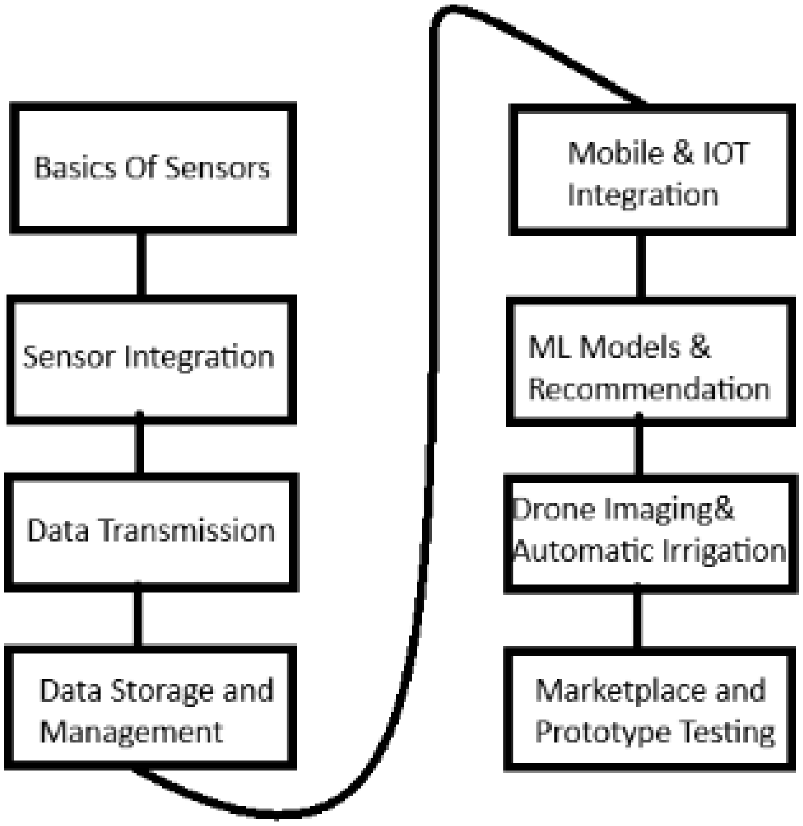

Basics Of Sensors
Sensor Integration
Data Transmission
Data Storage and Management
Mobile & IOT Integration
ML Models & Recommendation
Drone Imaging& Automatic Irrigation
Marketplace and Prototype Testing

Fig 4.4: Methodology Of the Whole Project

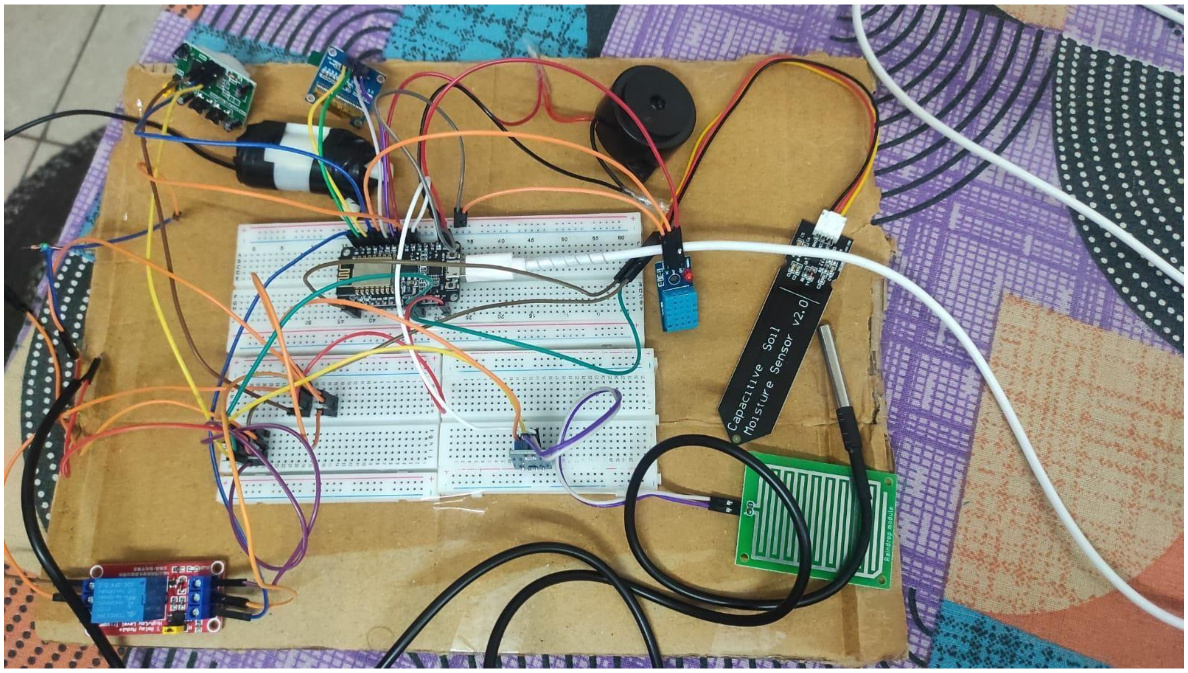


Fig 4.5: Completed Model of the System

Chapter 5
Experimentational Results

HARDWARE

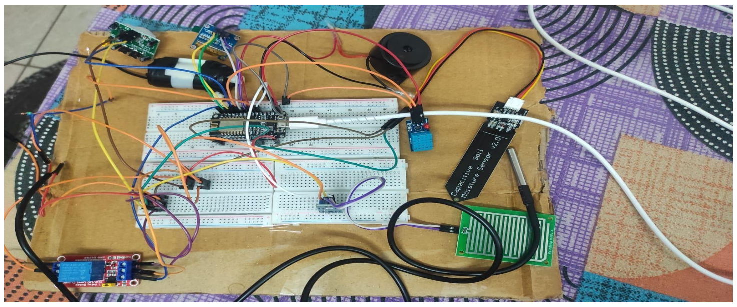


Fig 5.1: Compeleted Hardware Connections

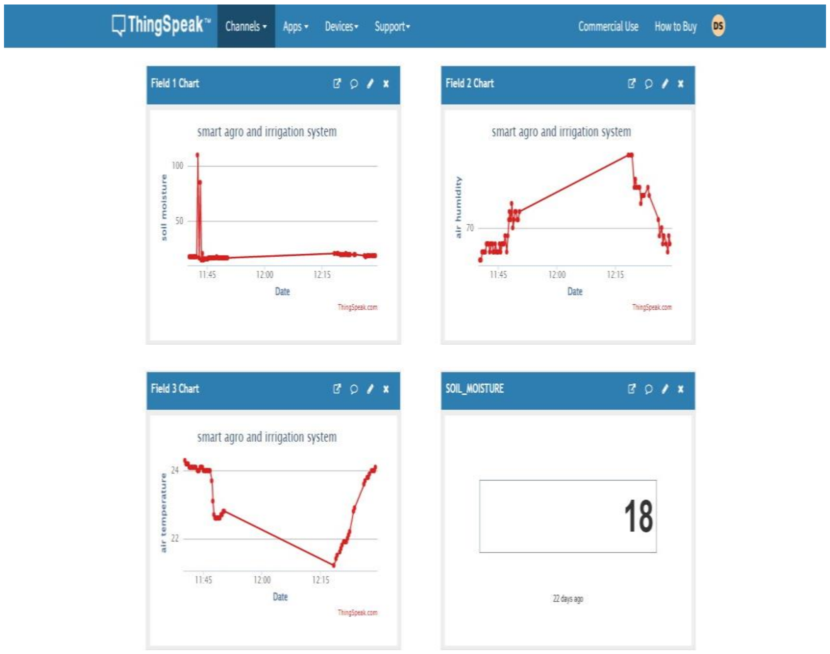


Fig 5.2: Data On ThinkSpeak

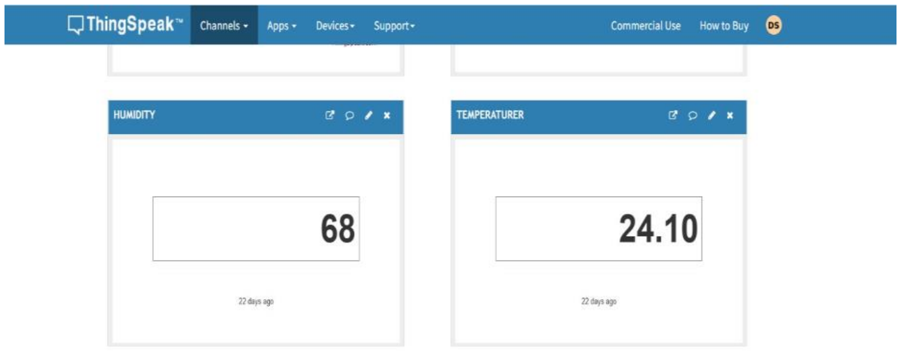


Fig 5.3: Some data analysis from Thinkspeak

The images vividly illustrate the seamless flow of data acquisition from various sensors, portraying the intricacies of how this information is subsequently transmitted and presented on the cloud server.

SOFTWARE:

The following images showcase the development of dedicated models for anomaly detection, crop prediction, and fertilizer prediction. Additionally, a detailed presentation of the user interface of the application is provided, offering a comprehensive view of the project's technical aspects.

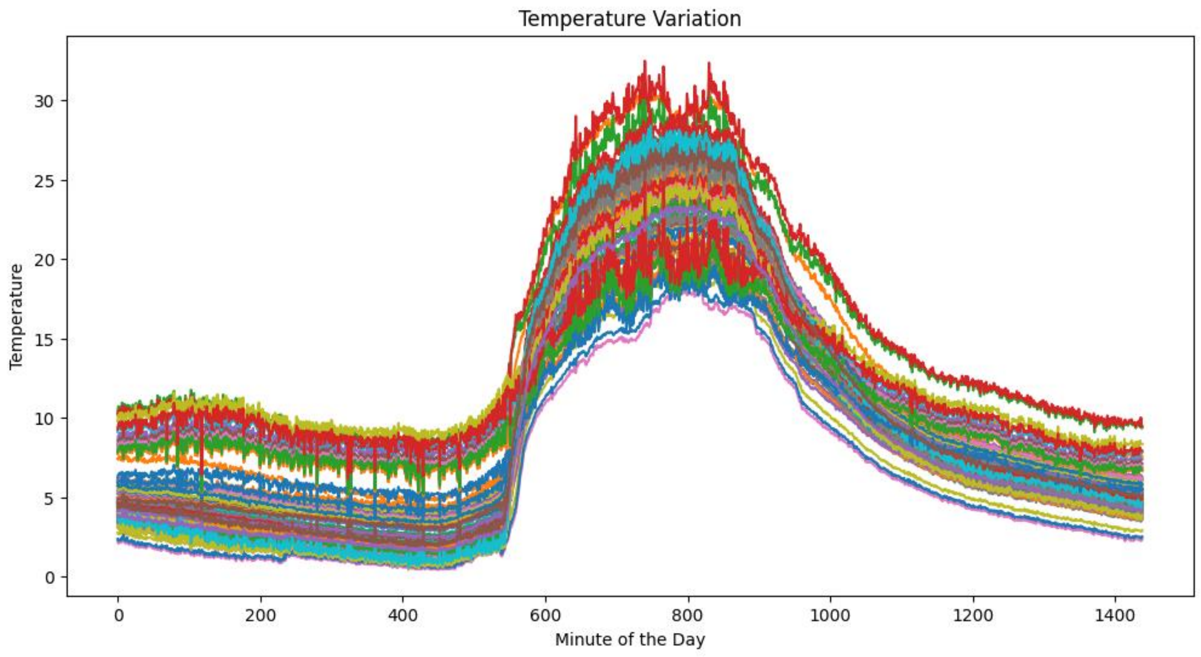


Fig 5.4: Input data to check temperature Variation

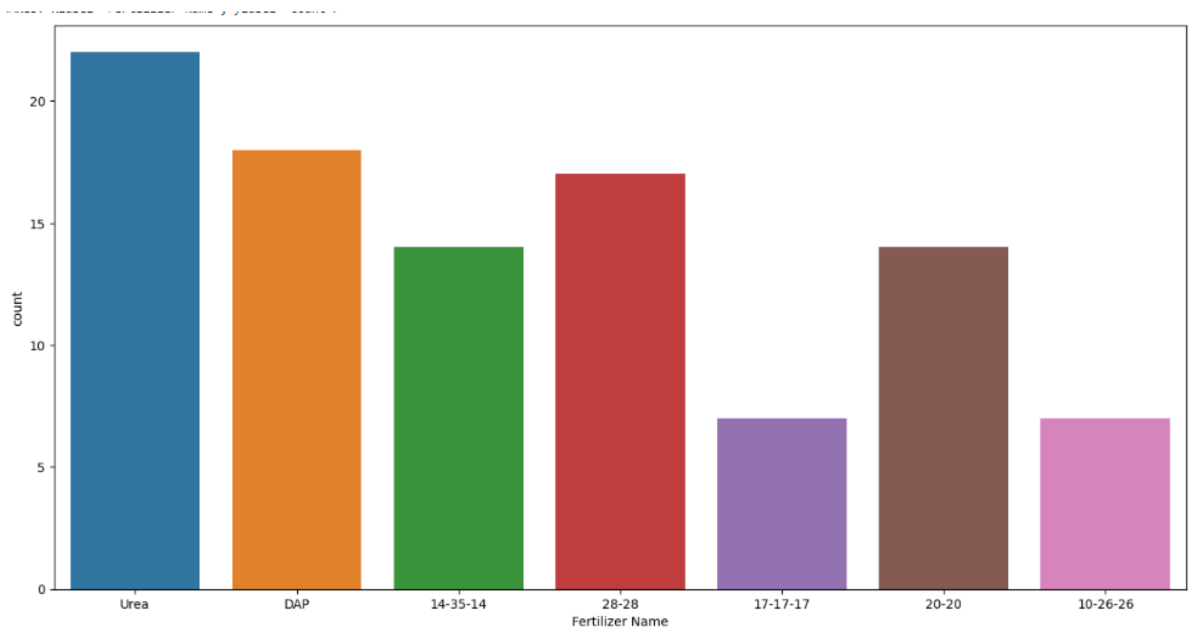
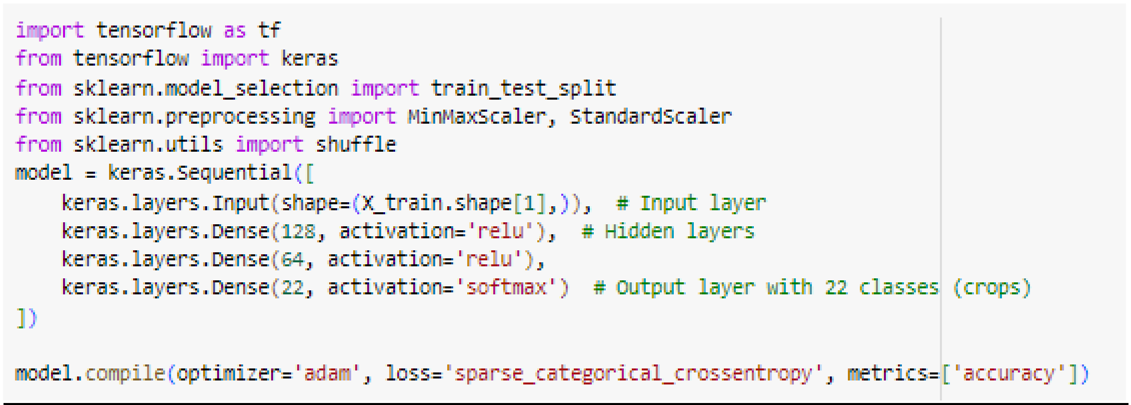


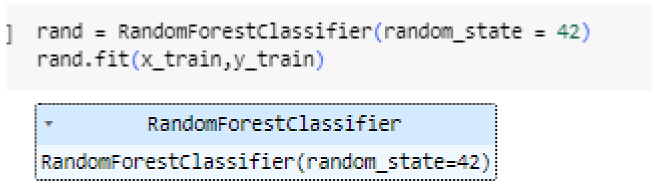
Fig 5.5: Fertilizer Distribution in Fretilizer Recommendation Model



```
import tensorflow as tf
from tensorflow import keras
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler, StandardScaler
from sklearn.utils import shuffle
model = keras.Sequential([
    keras.layers.Input(shape=(X_train.shape[1],)),  # Input layer
    keras.layers.Dense(128, activation='relu'),  # Hidden layers
    keras.layers.Dense(64, activation='relu'),
    keras.layers.Dense(22, activation='softmax')  # Output layer with 22 classes (crops)
])

model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
```

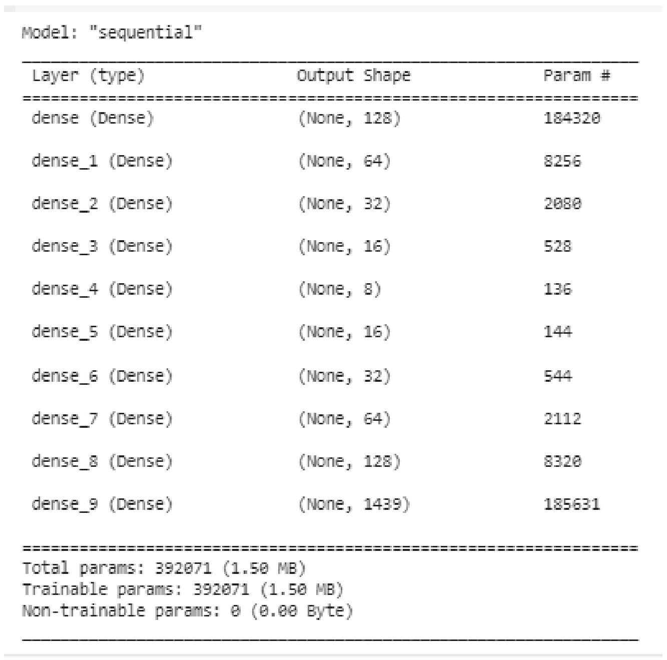
Fig 5.6: Crop Prediction Model and metrics used



```
] rand = RandomForestClassifier(random_state = 42)
  rand.fit(x_train,y_train)
```

```
RandomForestClassifier
RandomForestClassifier(random_state=42)
```

Fig 5.7: Fertilizer Prediction Model Used



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 128)               184320

 dense_1 (Dense)             (None, 64)                8256

 dense_2 (Dense)             (None, 32)                2080

 dense_3 (Dense)             (None, 16)                528

 dense_4 (Dense)             (None, 8)                 136

 dense_5 (Dense)             (None, 16)                144

 dense_6 (Dense)             (None, 32)                544

 dense_7 (Dense)             (None, 64)                2112

 dense_8 (Dense)             (None, 128)               8320

 dense_9 (Dense)             (None, 1439)              185631

=================================================================
Total params: 392071 (1.50 MB)
Trainable params: 392071 (1.50 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Fig 5.8: Model Architecture for Sensor Anomaly Detection

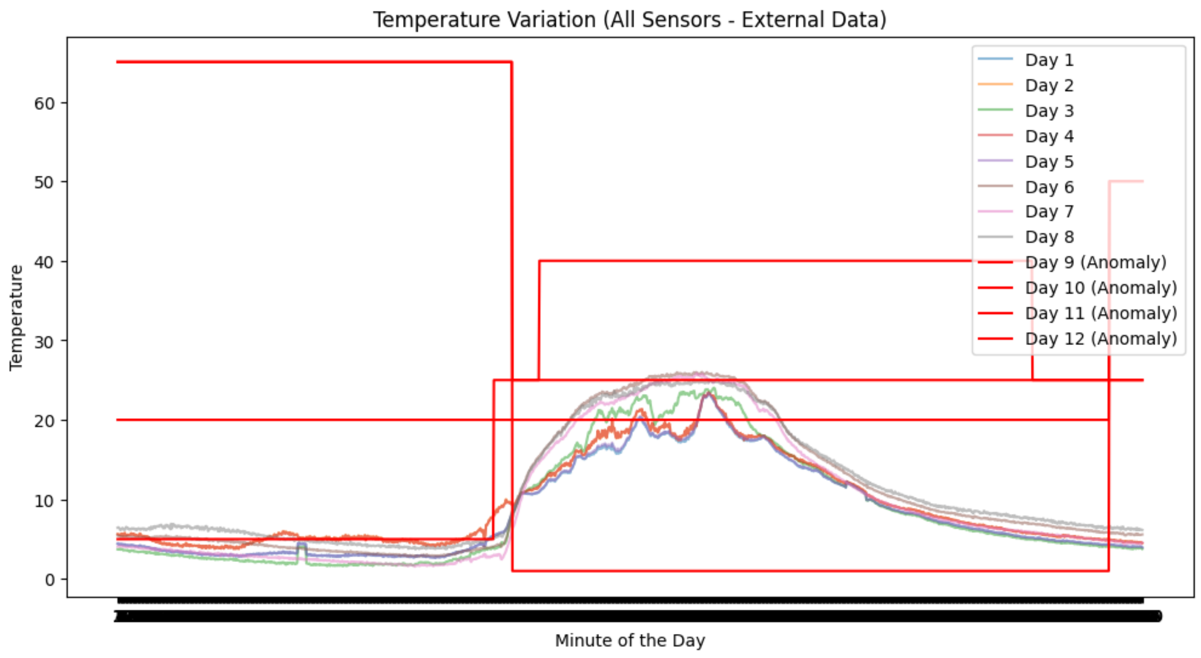


Fig 5.9: Anomaly Detection

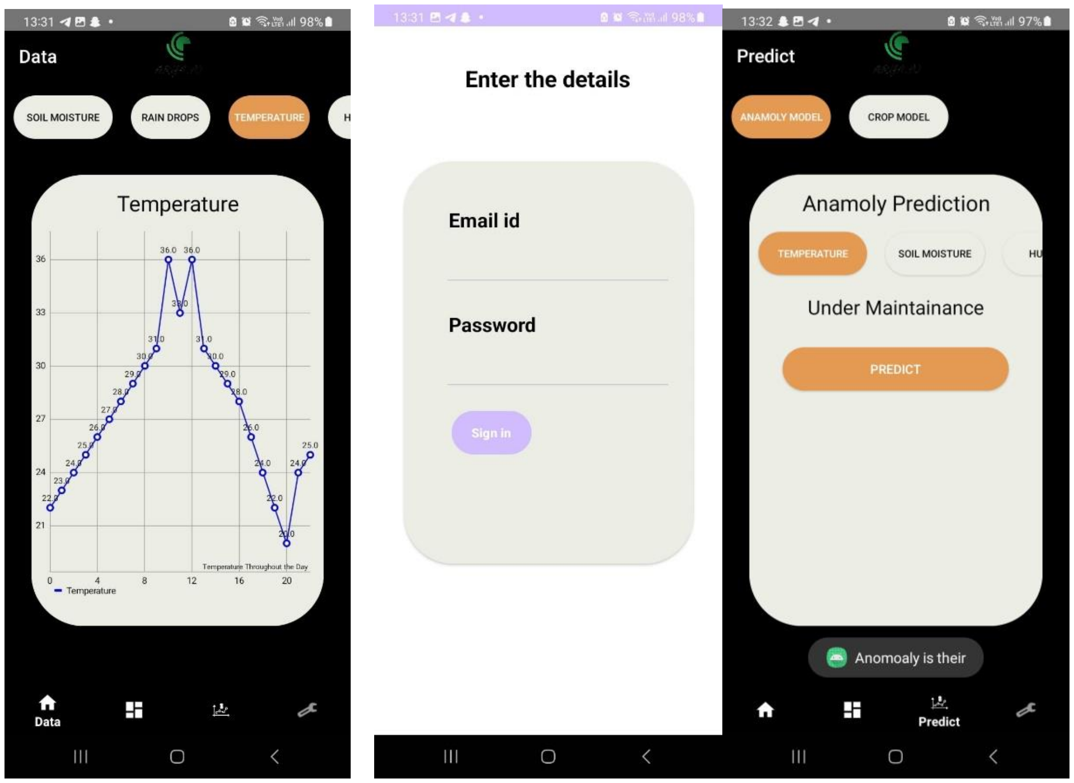


Fig 5.10: Application Design Layout

Chapter 5
Conclusions and Future Work

5.1 Conclusion

In conclusion, our project successfully addresses the multifaceted challenges in modern agriculture through the integration of IoT, machine learning, and mobile application development. The utilization of Autoencoders for sensor anomaly detection, deep learning neural networks for crop and fertilizer predictions, and the seamless integration of Firebase and MongoDB as backend databases contribute to the creation of a comprehensive and efficient smart agriculture system.

The implementation of an automatic irrigation system, real-time monitoring through a user-friendly Android app, and the incorporation of Winsorization techniques for outlier removal enhance the precision and sustainability of farming practices. The synergy of hardware and software components empowers farmers with data-driven insights, promoting resource optimization and informed decision-making.

5.2 Future Scope

Below are some of the future scopes or potential that this project can have.

- Explore advanced machine learning and deep learning models for further improving accuracy in crop and fertilizer predictions.
- Incorporate additional sensors for monitoring a broader range of environmental factors to allow more comprehensive data analysis.
- Implement predictive maintenance models for agricultural equipment, enhancing efficiency and longevity.
- Investigate edge computing solutions for real-time data processing, reducing dependency on cloud services and minimizing latency.
- Collaborate with agricultural experts and organizations to fine-tune machine learning models based on regional and crop-specific nuances.
- Enhance the mobile application with features for more detailed user interaction, including personalized insights and historical trends.
- Integrate sustainable agricultural practices, such as water conservation techniques, organic farming methodologies, and eco-friendly pest control strategies.
- Continuously evolve and refine the smart agriculture system to remain adaptable to diverse farming scenarios and technological advancements.

References

1) K. Taneja and S. Bhatia, "Automatic irrigation system using Arduino UNO," 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), 2017, pp. 132-135, doi: 10.1109/ICCONS.2017.8250693.

2) Monica, M., Yeshika, B., Abhishek, G. S., Sanjay, H. A., & Dasiga, S. (2017, October). IoT based control and automation of smart irrigation system: An automated irrigation system using sensors, GSM, Bluetooth and cloud technology. In 2017 International Conference on recent innovations in signal processing and embedded systems (RISE) (pp. 601-607). IEEE.

3) Singh, P., & Saikia, S. (2016, December). Arduino-based smart irrigation using water flow sensor, soil moisture sensor, temperature sensor and ESP8266 WiFi module. In 2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC) (pp. 1-4). IEEE.

4) Yasin, H. M., Zeebaree, S. R., & Zebari, I. M. (2019, April). Arduino based automatic irrigation system: Monitoring and SMS controlling. In 2019 4th Scientific International Conference Najaf (SICN) (pp. 109-114). IEEE.

5) Serdaroglu, K. C., Onel, C., & Baydere, S. (2020, December). IoT Based Smart Plant Irrigation System with Enhanced learning. In 2020 IEEE Computing, Communications and IoT Applications (ComComAp) (pp. 1-6). IEEE.

Appendices

Appendix A

Code Attachments

A.1 Arduino Code

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266HTTPClient.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1

#define PIR_PIN D3
#define RAIN_PIN D7
#define DHTPIN D4
#define BUZZER_PIN D5

#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <DHT.h>

DHT dht(DHTPIN, DHT11);

String apiKey = "DJHSWF9Y7OX2PE17";
const char *ssid = "Redmi Note 9 Pro Max";
const char *pass = "divyansh123";
const char *server = "192.168.28.141";  // Replace with the IP address of your Node.js server
const int serverPort = 3000;

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

const int AirValue = 790;
const int WaterValue = 390;
const int SensorPin = A0;

int soilMoistureValue = 0;
int soilmoisturepercent = 0;

WiFiClient client;

void setup() {
  Serial.begin(115200);
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.clearDisplay();
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(PIR_PIN, INPUT);
  pinMode(RAIN_PIN, INPUT);
  dht.begin();

  WiFi.begin(ssid, pass);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  delay(4000);
}

void loop() {
  int motion = digitalRead(PIR_PIN);
  int rain = analogRead(RAIN_PIN);

  if (motion == 1) {
    Serial.println("Motion detected");
  } else {
    Serial.println("Motion not detected!");
  }

  if (rain == 0) {
    Serial.println("Rain detected");
  } else {
    Serial.println("Rain not detected");
  }

  float h = dht.readHumidity();
  float t = dht.readTemperature();

  Serial.print("Humidity: ");
  Serial.println(h);
  Serial.print("Temperature: ");
  Serial.println(t);

  soilMoistureValue = analogRead(SensorPin);
  Serial.println(soilMoistureValue);

  soilmoisturepercent = map(soilMoistureValue, AirValue, WaterValue, 0, 100);
```

```
if (soilmoisturepercent > 100) {
  Serial.println("100 %");

  display.setCursor(0, 0);
  display.setTextSize(2);
  display.setTextColor(WHITE);
  display.print("Soil RH:");
  display.setTextSize(1);
  display.print("100");
  display.println(" %");
  display.setCursor(0, 20);
  display.setTextSize(2);
  display.print("Air RH:");
  display.setTextSize(1);
  display.print(h);
  display.println(" %");
  display.setCursor(0, 40);
  display.setTextSize(2);
  display.print("Temp:");
  display.setTextSize(1);
  display.print(t);
  display.println(" C");
  display.display();

  delay(250);
  display.clearDisplay();
}

else if (soilmoisturepercent < 0) {
  Serial.println("0 %");

  display.setCursor(0, 0);
  display.setTextSize(2);
  display.setTextColor(WHITE);
  display.print("Soil RH:");
  display.setTextSize(1);
  display.print("0");
  display.println(" %");
  display.setCursor(0, 20);
  display.setTextSize(2);
  display.print("Air RH:");
  display.setTextSize(1);
  display.print(h);
  display.println(" %");
  display.setCursor(0, 40);
  display.setTextSize(2);
  display.print("Temp:");
  display.setTextSize(1);
  display.print(t);
  display.println(" C");
  display.display();

  delay(250);
  display.clearDisplay();
}

else if (soilmoisturepercent >= 0 && soilmoisturepercent <= 100) {
  Serial.print(soilmoisturepercent);
  Serial.println("%");

  display.setCursor(0, 0);
  display.setTextSize(2);
  display.setTextColor(WHITE);
  display.print("Moisture:");
```

```
    display.setTextSize(1);
    display.print(soilmoisturepercent);
    display.println(" %");
    display.setCursor(0, 20);
    display.setTextSize(2);
    display.print("Air RH:");
    display.setTextSize(1);
    display.print(h);
    display.println(" %");
    display.setCursor(0, 40);
    display.setTextSize(2);
    display.print("Temp:");
    display.setTextSize(1);
    display.print(t);
    display.println(" C");
    display.display();

    delay(250);
    display.clearDisplay();
  }

  if (soilmoisturepercent >= 0 && soilmoisturepercent <= 30) {
    digitalWrite(BUZZER_PIN, HIGH);
    Serial.println("Motor is ON");
  } else if (soilmoisturepercent > 30 && soilmoisturepercent <= 100) {
    digitalWrite(BUZZER_PIN, LOW);
    Serial.println("Motor is OFF");
  }

 // ... (previous code)

if (client.connect(server, serverPort)) {
    String postPath = "/data/id=650156153c813771bfbe06e7";
    String postContent = "{"
                  "\"_id\":\"650156153c813771bfbe06e7\","
                  "\"Soilmoisture\":\"" + String(soilmoisturepercent) + "\","
                  "\"Raindrops\":\"" + String(rain == 0 ? "1" : "0") + "\","
                  "\"Temperature\":\"" + String(t) + "\","
                  "\"Humidity\":\"" + String(h) + "\","
                  "\"Waterpump\":\"" + String(soilmoisturepercent > 30 ? "0" : "1") + "\","
                  "\"__v\":0"
                  "}";

    client.print("PUT " + postPath + " HTTP/1.1\r\n");
    client.print("Host: " + String(server) + "\r\n");
    client.print("Content-Length: " + String(postContent.length()) + "\r\n");
    client.print("Content-Type: application/json\r\n\r\n");
    client.print(postContent);

    Serial.println("Request sent");

    // Wait for response
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        Serial.print(c);
      }
    }

    Serial.println("Request complete");

    client.stop();
} else {
    Serial.println("Error: Unable to connect to the server");
```

```
}

// ... (rest of the code)


  delay(5000); // Wait for 30 seconds before the next upload
}
```

MODEL CODE :

https://colab.research.google.com/drive/1zU0c7LjOkUlA1m7ZAgsZokkciA0wRolU?usp=sharing

Android Code:

https://github.com/rishav1107/Sensor_Anomaly_Detection_and_Data_Imputation_using-_Autoencoders