CHAPTER 1

INTRODUCTION

1.1 Motivation

Agriculture has been the backbone of human civilization since ancient times. Today, as global populations surge and urbanization intensifies, the demand for food production is rising sharply. However, traditional farming methods struggle to meet these growing needs due to resource scarcity, labour shortages, climate variability, and environmental degradation. There is a pressing need for innovative, sustainable, and efficient farming practices.

Smart Farming emerges as a transformative solution. By harnessing advanced technologies like the Internet of Things (IoT), Artificial Intelligence (AI), and Machine Learning (ML), smart farming empowers farmers to make data-driven decisions, optimize resource utilization, enhance crop yields, and reduce environmental impact.

The motivation behind this project stems from the desire to build a cost-effective, scalable, and intelligent system that empowers small and medium-scale farmers by providing real-time monitoring, predictive analytics, and automation capabilities without complex or expensive infrastructure.

1.2 Existing Models

Several existing smart farming solutions have been developed in recent years:

- Basic IoT Monitoring Systems: These systems use sensors to collect soil moisture, temperature, and humidity data. Data is transmitted to a server where it is displayed to the user. However, these models often lack intelligent decision-making and rely on the farmer's judgment.
- Standalone Crop Recommendation Systems: Some solutions focus solely on crop selection based on soil properties but do not integrate real-time field data or provide automation features.
- Expensive Proprietary Platforms: Large companies offer comprehensive smart agriculture platforms, but these are often cost-prohibitive for small farmers.

- Manual Control with Data Feedback: In some systems, farmers receive sensor data on a mobile app but must manually control irrigation systems or fertilizer applications.

Limitations of Existing Models

- Lack of integrated AI-based automation.
- High setup and maintenance costs.
- Limited adaptability for small and medium farms.
- Insufficient real-time predictive capabilities.

1.3 Proposed Model

The proposed system aims to bridge the gap by integrating IoT, AI, and automation into a cohesive, affordable framework.

Key components include:

1. **IoT-based Data Acquisition:**
    - Sensors for soil pH, temperature, and moisture.
    - Wireless communication via ESP32 Wi-Fi microcontroller.
2. **AI/ML-based Decision Support:**
    - Data preprocessing using AI techniques.
    - ML algorithms (Linear Regression, Decision Tree, and Random Forest) for crop recommendation.
    - Random Forest chosen as the final model due to its robustness and high accuracy.
3. **Automation and Actuation:**
    - Automated control of pH levels and irrigation using motor controllers based on sensor data and AI recommendations.
4. **Deployment via Flask:**
    - A lightweight, easy-to-deploy backend server using Flask for model inference and control logic.

This holistic system not only provides real-time monitoring but also actionable intelligence and field actuation, minimizing human intervention.

1.4 Problem Statement

"How can we design a scalable, affordable smart farming system that utilizes IoT and AI/ML to recommend suitable crops, optimize resource usage, and automate irrigation, thereby increasing farm productivity and sustainability?"

This project aims to develop such a system with an emphasis on real-time field monitoring, AI-driven decision-making, and automatic actuation using cost-effective hardware.

1.5 Objectives

The main objectives of this project are:

- To develop an IoT-enabled platform that collects real-time field parameters like pH, soil moisture, and temperature.
- To preprocess collected data and build a predictive model using AI/ML techniques.
- To recommend crops suitable for the current field conditions.
- To automate irrigation and nutrient management using motor controllers based on AI predictions.
- To create a Flask-based server for model deployment and sensor communication.

1.6 Challenges

Despite the promising potential, the project faces several technical and practical challenges:

- Sensor Calibration and Reliability: Ensuring accurate readings in diverse environmental conditions.
- Data Preprocessing: Handling missing, noisy, or inconsistent sensor data.
- Real-time Data Transmission: Ensuring stable and secure wireless communication in remote fields.
- Integration Complexity: Seamlessly integrating hardware, AI models, and automation controls.

- Power Management: Minimizing power consumption for continuous operation in off-grid areas.
- User Accessibility: Designing a user interface simple enough for non-technical users.

1.7 Scope

This project is intended primarily for small to medium-sized farms, focusing on:

- Soil-based crop recommendation using AI/ML.
- Real-time monitoring of environmental conditions.
- Automated water and pH control for soil optimization.
- Scalable architecture that can accommodate additional sensors or actuators.
- Deployment in rural or semi-urban settings where traditional farming is predominant.

While the current version focuses on basic soil and crop optimization, future extensions could include:

- Integration with weather forecast APIs.
- Drone-based imaging for crop health monitoring.
- Fertilizer and pesticide recommendation systems.
- Edge AI deployment for offline operation.

CHAPTER 2

LITERATURE SURVEY

2.1 Introduction to Smart Farming and Precision Agriculture

The agricultural sector has been revolutionized by technological innovations in recent decades. Traditional farming methods, though reliable historically, have become insufficient to meet the needs of the modern world, characterized by rapid population growth, resource scarcity, and environmental challenges. Smart farming and precision agriculture have emerged as technological interventions aimed at addressing these issues.

Smart farming leverages advanced technologies such as the Internet of Things (IoT), Machine Learning (ML), Artificial Intelligence (AI), Big Data, and cloud computing to automate and optimize agricultural operations. Precision agriculture, a subset of smart farming, focuses on the precise management of crops and resources at a micro-level, ensuring maximum efficiency and minimum waste.

Together, these approaches aim to increase agricultural productivity, ensure sustainability, reduce environmental impacts, and improve decision-making processes in farming.

2.2 IoT in Smart Farming

The Internet of Things (IoT) has introduced significant transformations in agriculture. IoT devices, embedded with sensors and connectivity features, enable the real-time collection of data on critical farming parameters such as soil moisture, temperature, humidity, crop health, and livestock conditions.

The primary advantages of IoT integration include:

- Real-time Monitoring: Farmers can track environmental and crop conditions continuously.
- Remote Management: IoT devices allow control of irrigation systems, fertilization schedules, and pest control mechanisms remotely.
- Precision Resource Usage: Data from IoT sensors enable precise water and fertilizer application, minimizing resource wastage and promoting sustainability.

Furthermore, cloud platforms often store the sensor data, allowing advanced analytics and machine learning models to generate actionable insights. Mobile and web applications facilitate easy access to these insights, enhancing the convenience for farmers.

2.3 LoRaWAN: The Backbone of Smart Agricultural Connectivity

A significant challenge in deploying IoT devices in agriculture is ensuring reliable communication across vast, often remote, agricultural fields. Traditional Wi-Fi or cellular networks are not always viable options due to range limitations and power consumption concerns.

The Long-Range Wide Area Network (LoRaWAN) protocol emerges as a compelling solution:

- Long Range: LoRaWAN can connect devices across several kilometers.
- Low Power Consumption: IoT sensors using LoRaWAN can operate for years without frequent battery changes, making the technology sustainable.
- Cost-Effective Deployment: Operating in unlicensed frequency bands reduces the operational cost, making it suitable for large-scale farms.

LoRaWAN enables seamless, scalable, and reliable communication between a multitude of farm-deployed devices and central data hubs. Case studies in vineyards and livestock monitoring demonstrate the effectiveness of LoRaWAN-enabled smart farming systems.

2.4. Machine Learning and Anomaly Detection in Agriculture

The integration of machine learning algorithms into smart farming platforms has unlocked new capabilities in data analysis, prediction, and anomaly detection.

- Isolation Forest: This unsupervised machine learning method is particularly effective for anomaly detection. In smart farming, it identifies irregularities in environmental parameters such as soil moisture or crop health, enabling early intervention.
- Linear Regression: Used to predict future values (e.g., soil moisture levels, yield estimates) based on historical sensor data.
- Random Forest: An ensemble learning method that enhances prediction accuracy for outcomes like crop yields, pest outbreaks, and livestock health indicators.

These predictive models improve the farmers' ability to anticipate and mitigate risks, optimize resource utilization, and enhance overall productivity.

2.5. Role of Big Data, Deep Learning, and Blockchain

In addition to conventional machine learning, the advent of big data analytics and deep learning models has further enhanced precision agriculture.

- Deep Learning (DL): By analyzing large datasets (e.g., images from drones), DL models can identify crop diseases early, predict plant growth stages, and suggest interventions.
- Big Data Analytics: Aggregated environmental and operational data allow trend analysis, resource optimization, and strategic planning.
- Blockchain Integration: Ensures traceability, transparency, and trust across the agricultural supply chain by recording every farming process immutably.

The combination of these technologies elevates smart farming from mere automation to intelligent decision support and supply chain optimization.

2.6. Challenges in Smart Farming

Despite its potential, smart farming faces several significant challenges:

- Data Quality and Availability: Inconsistent or insufficient data can compromise the effectiveness of AI and ML models.
- High Initial Investment: Setting up IoT-based farming systems involves substantial costs, making it challenging for small-scale farmers.
- Privacy and Security: Farmers' data must be protected against unauthorized access and misuse.
- Skill Gaps: Farmers need training to effectively utilize these advanced systems.
- Environmental Factors: Unpredictable weather patterns can still disrupt even the best technology-enabled plans.

Addressing these challenges is crucial to achieving the widespread adoption and success of smart farming technologies.

2.7 Case Studies and Implementations

Several real-world implementations validate the effectiveness of smart farming systems:

- LoRaFarM: A modular IoT architecture for smart farming based on LoRaWAN demonstrated real-time environmental monitoring and data visualization for vineyards and greenhouse crops.
- Aquaculture Monitoring: Machine learning techniques (Isolation Forest, K-Means) applied to water quality datasets have improved fish farm management.
- Automated Fertilizer and Irrigation Systems: Using wireless sensors and drones for targeted delivery has significantly reduced resource use while improving yields.

These implementations confirm that the integration of IoT, LoRaWAN, and AI enhances farm productivity, resource efficiency, and environmental sustainability.

Table 2.7: Summary of Research Papers

No	Title	Authors & Year	Key Focus	Methods/ Technologies Used	Findings	Relevance to Smart Agriculture
1	Evaluation of Machine Learning Approaches for Precision Farming in Smart Agriculture Systems	Ghulam Mohyuddin et al., 2024	Comprehensive review of ML in precision agriculture	ML algorithms (SVM, CNN, RF), IoT, UAVs	ML enhances soil analysis, crop prediction, disease detection, and irrigation control	Demonstrates ML's critical role in transforming traditional farming into data-driven, efficient practices
2	Robust and Efficient Machine Learning Model for Smart Farming Decisions	S.K. Apat et al., 2022	Development of a Heterogeneous Ensemble Learning Environment (HELE)	Random Forest, SVM, KNN, LSTM, IoT sensor data	HELE outperforms single models in yield prediction (up to 99.27% accuracy)	Highlights benefit of ensemble ML systems using real-time environmental data for

						smart decisions
3	LoRaFarM: A LoRaWAN-Based Smart Farming Modular IoT Architecture	Codeluppi et al., 2020	Modular IoT system for data collection and monitoring	LoRaWAN, WSNs, cloud-based dashboards	Deployed in real farms for soil and crop monitoring; energy-efficient and scalable	Offers low-cost, scalable architecture for integrating IoT into smart farms
4	Intelligent LoRaWAN-Based IoT Device for Monitoring & Control in Smart Farming	Alumfareh et al., 2024	Smart farming via ML-embedded IoT system	LoRaWAN, Isolation Forest, RF, LR	Real-time anomaly detection and prediction of temperature/humidity variations	Enables precision agriculture in remote, resource-limited farms
5	Deep Learning Agricultural Information Classification with IoT	He Yang et al., 2022	Use of DL for scientific content classification in agriculture	DL (forward/reverse algorithms), IoT, cross-entropy optimization	Achieved high classification accuracy; improved search relevance	Enhances agricultural research dissemination and supports economic management.
6	Role of Sensors, Big Data & ML in Modern Animal Farming	S. Neethirajan, 2020	Review of digital technologies in livestock farming	Sensors (RFID, thermal, vision), ML, big data, mechanistic models	Boosts productivity, welfare, and disease prevention in livestock	Critical for transitioning animal farming to tech-enhanced, scalable operations

7	Smart Farming and Precision Agriculture: Need in Today's World	Sreya John & P.J. Arul Leena Rose, 2024	Conceptual analysis of smart and precision farming	IoT, ML, Big Data, automation systems	Enhances food security, reduces environmental impact, optimizes input use	Establishes smart farming as essential for sustainable global agriculture

2.8 How the Proposed Model Addresses the Gaps

Table 2.8: Identified Gaps and its Solutions

Identified Gap	**Proposed Solution**
Limited AI Integration	Real-time crop recommendation using ML models.
Manual Irrigation Control	Automatic motor-driven irrigation system.
High Cost	Use of affordable ESP32 and basic sensors.
Need for Scalability	Modular system design for easy expansion.
Internet Dependency	Local Flask server deployment; optional cloud backup.

CHAPTER 3

DOMAIN ANALYSIS

3.1 The Internet of Things (IoT) in Agriculture

3.1.1 Introduction to IoT

The **Internet of Things (IoT)** refers to a network of interconnected physical devices embedded with sensors, software, and connectivity, allowing them to collect and exchange data.
In agriculture, IoT enables farmers to remotely monitor field conditions and automate processes, leading to **smart farming** or **precision agriculture**.

3.1.2 Role of IoT in Smart Farming

In the context of farming, IoT devices offer:

- **Real-time Data Collection**: Soil moisture, temperature, humidity, and pH values are continuously monitored.
- **Remote Monitoring**: Farmers can observe farm conditions without physical presence.
- **Automated Control**: Motors, sprinklers, and pH controllers are activated automatically based on sensor readings.
- **Predictive Insights**: Data collected over time can reveal patterns and trends critical for decision-making.

3.1.3 Components of IoT System in Proposed Model

Table 3.1.3.1: Components of IOT

Component	Description
ESP32 36-pin Wi-Fi Module	Core controller handling sensor data collection and transmission.
pH Sensor	Measures soil acidity or alkalinity.
Temperature Sensor	Monitors ambient temperature affecting crop growth.

Soil Moisture Sensor	Measures soil water content to optimize irrigation.
Motor Controllers	Actuate irrigation pumps based on sensor input.
Wi-Fi Network	Enables wireless communication between devices and server.

3.1.4 Architecture Overview

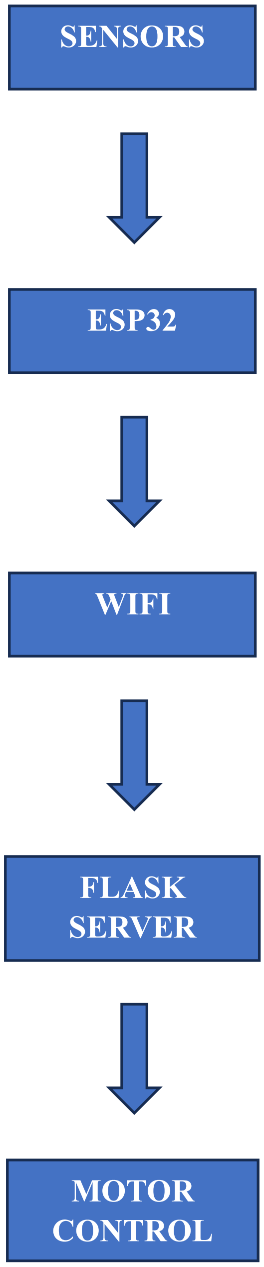

Figure 3.1.4.1: Architecture Overview

3.1.5 Benefits of IoT in Agriculture

- **Efficiency**: Reduced resource wastage (water, fertilizers).
- **Productivity**: Higher crop yield through better decision-making.
- **Cost-effectiveness**: Lower labor costs and minimal field visits.
- **Scalability**: Easy addition of new sensors or modules.

3.1.6 Challenges in IoT Deployment

- **Network Connectivity**: Rural areas often have weak Wi-Fi or cellular coverage.
- **Power Consumption**: Maintaining sensor activity in remote fields requires sustainable power solutions (e.g., solar panels).
- **Sensor Calibration**: Accuracy degrades over time due to environmental exposure.
- **Data Security**: Protecting farm data from unauthorized access is critical.

3.2 Cloud Computing in Agriculture

3.2.1 Introduction to Cloud Computing

Cloud computing refers to the delivery of computing services—including servers, storage, databases, networking, software, and analytics—over the Internet ("the cloud"). It enables on-demand access to large-scale computational power and storage, without investing in physical hardware.

3.2.2 Role of Cloud in Smart Farming

In Smart Farming:

- **Data Storage**: Storing massive amounts of real-time sensor data for future analysis.
- **Advanced Analytics**: Leveraging cloud-based AI/ML services for crop prediction models.
- **Remote Access**: Farmers can view field data from anywhere, anytime.
- **Backup and Recovery**: Ensures important farm data is safe even in case of device failure.

3.2.3 Integration in Proposed System

Although the proposed model primarily uses a **local Flask server** for model deployment (to minimize dependency on internet availability), it can optionally:

- Push sensor data to a cloud database (e.g., Firebase, AWS DynamoDB).
- Perform periodic cloud backups.
- Use cloud-hosted dashboards for larger farms.

3.2.4 Advantages of Cloud Computing in Smart Farming

Table 3.2.4.1: Benefits of Cloud Computing

Benefit	Description
Scalability	Easily accommodates increasing number of devices and data.
Flexibility	Access services and data from anywhere with an internet connection.
Cost Efficiency	Pay-as-you-go models avoid large initial investments.
Advanced Tools	Access to AI, Big Data, and real-time analytics services.

3.2.5 Challenges in Cloud Adoption

- **Internet Dependence**: Requires continuous, stable connectivity.
- **Data Privacy**: Sensitive farm data could be vulnerable to breaches.
- **Vendor Lock-in**: Migration between cloud providers can be complex and costly.
- **Skill Requirement**: Farmers may require training to manage cloud services.

3.3 Combined Impact of IoT and Cloud in Smart Farming

By combining **IoT for real-time field sensing** and **Cloud Computing for advanced analytics and storage**, farmers are empowered to:

- Receive predictive alerts (e.g., impending drought conditions).
- Optimize irrigation schedules automatically.
- Select best crops dynamically based on field health trends.
- Maintain historical records of soil conditions and farming outcomes.

The synergy between IoT and Cloud thus forms the backbone of a **sustainable, intelligent, and high-yield farming ecosystem**.

CHAPTER 4

METHODOLOGY

4.1 Overview of the System

The proposed Smart Farming system integrates **IoT devices** (ESP32 + sensors), **Machine Learning models** (Random Forest), and **automated actuator control** to monitor, predict, and optimize crop production.

The architecture consists of:

- **Data Collection Layer** (Sensors + ESP32)
- **Data Processing Layer** (Flask Server + ML Models)
- **Decision Making Layer** (Crop Recommendation + Irrigation Control)
- **Action Layer** (Motor control for water supply and pH controller)

4.2 System Architecture

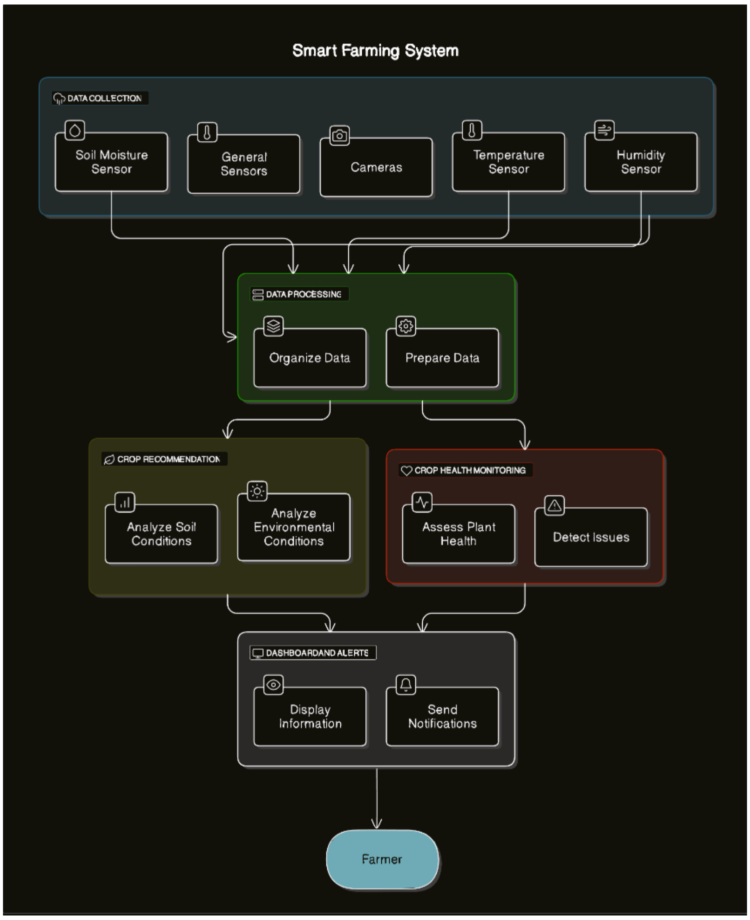

Figure 4.2.1: System Architecture.

4.3 Step-by-Step Working

4.3.1 Data Acquisition Layer (IoT Sensors)

Table 4.3.1.1: Data Acquisition Devices

Device	**Function**
Soil Moisture Sensor	Measures soil water content.
Temperature Sensor	Monitors environmental temperature.
pH Sensor	Measures soil pH (acidity/alkalinity).
ESP32 Microcontroller	Collects sensor data and transmits wirelessly.

Process:

- Sensors are deployed across the farm field.
- Readings are captured periodically (e.g., every 10 minutes).
- ESP32 consolidates sensor data and sends it over Wi-Fi.

4.3.2 Data Transmission Layer

- **Protocol**: Data is transmitted via Wi-Fi using HTTP requests.
- **Server**: A lightweight **Flask server** running on a local or cloud machine receives the data.

4.3.3 Data Preprocessing Layer (Flask + AI/ML)

Technologies Used:

- **Python** (for backend server and model)
- **Flask** (for API and data handling)
- **Scikit-learn** (for machine learning)

Steps:

1. **Data Cleaning**: Remove missing, corrupted, or extreme values.
2. **Feature Scaling**: Normalize sensor values to improve ML performance.
3. **Feature Engineering**: Generate derived features if necessary (e.g., average soil moisture, day-wise variations).

4.3.4 Machine Learning Model Layer (Crop Recommendation)

Table 4.3.4.1: Machine Learning Algorithms Used

Algorithm	**Role**
Linear Regression	For preliminary analysis (not final model).
Decision Tree	For early experiments and comparison.
Random Forest	**Final model** used for best accuracy.

Random Forest Model:

A **Random Forest** is an **ensemble learning algorithm** that builds multiple **decision trees** and merges their results to obtain a more accurate and stable prediction. It belongs to the family of supervised learning algorithms and is particularly useful for classification and regression tasks in complex datasets — like those encountered in precision agriculture.

1. **Dataset Sampling:** The algorithm takes multiple random samples (with replacement) from the original dataset (this is known as bagging).
2. **Tree Construction:** For each sample, it builds a separate decision tree by selecting a random subset of features (e.g., soil pH, moisture, humidity, etc.).
3. **Prediction Aggregation:**
    - **Classification Tasks:** It takes a majority vote from all decision trees to decide the predicted class (e.g., wheat, rice, maize).
    - **Regression Tasks:** It takes the average of all decision trees' outputs.

Why Random Forest?

Table 4.3.4.2: Benefits of Random Forest Algorithm

Advantage	**Explanation**
Handles Noisy Data Well	Random sampling and averaging reduce the impact of outliers or sensor errors.
Reduces Overfitting	Individual decision trees may overfit, but combining many trees generalizes better.

Robust to Missing Data	Can still make reasonable predictions even with incomplete feature values.
High Accuracy	Studies report accuracy in the range of **90%–95%**, depending on the dataset quality and size.
Feature Importance Estimation	Highlights which parameters (e.g., soil pH vs. temperature) influence predictions most.

4.3.5 Decision Making Layer

After receiving sensor data:

- The ML model predicts the best crop to cultivate at that time.
- Based on soil moisture levels:
    - If below a certain threshold, irrigation motor is activated automatically.
- Based on pH levels:
    - If soil acidity is outside optimal range, pH controller is triggered.

Sample Thresholds:

Parameter	Threshold Value	Action
Soil Moisture	< 30%	Turn ON irrigation motor.
Soil pH	< 5.5 or > 7.5	Activate pH controller.
Temperature	> 38°C	Send alert for shading/cover.

4.3.6 Actuation Layer

Motor Controller:

- Receives ON/OFF commands based on decisions made by Flask server.

pH Controller:

- Adds corrective agents (acidic/basic solutions) when needed.

Water Motor:

- Supplies water for irrigation based on real-time soil moisture monitoring.

Manual Override:

- Farmer can override automated decisions through a mobile or web interface.

4.5 Key Innovations in the Proposed Methodology

Table 4.5.1: Summary of Innovations

Innovation	Description
Low-Cost IoT Deployment	Using ESP32 and basic sensors for affordability.
AI-Enabled Smart Decisions	Real-time crop recommendation based on environmental data.
Automated Field Actuation	Motors and pH controllers operate without manual effort.
Modular and Scalable Design	Easy addition of new sensors, actuators, or cloud modules.

CHAPTER 5

REQUIREMENT COLLECTION

5.1 Hardware Requirements

The proposed smart farming system requires specific hardware components to collect environmental data and actuate control systems. Below is the list of hardware needed:

Table 5.1.1: Hardware Requirements

Component	Specification	Purpose
ESP32 (36 Pin)	Wi-Fi enabled microcontroller	Central control unit for sensor integration and data transmission.
pH Sensor	0–14 pH Range, Analog Output	Measures the acidity or alkalinity of the soil.
Temperature Sensor (DHT22/DS18B20)	Temperature range: -40°C to +80°C	Measures the environmental temperature affecting crops.
Soil Moisture Sensor	Capacitive type preferred	Measures water content in soil.
Humidity Sensor	Included with DHT22 (optional)	Measures atmospheric humidity.
Motor Driver Module	L298N or similar	Controls ON/OFF states of motors.
Water Pump Motor	12V DC motor	Supplies water to crops based on soil moisture.
pH Controller Unit	Automated solution dispensing system	Adjusts soil pH by adding acidic/basic solution.
Power Supply	5V/12V adapters and regulators	Powering sensors and motors.

Connecting Wires	-	For sensor and module connections.
Breadboard or PCB	-	For circuit assembly and prototyping.

5.1.1 ESP 32

It is dual core MCU from Espressif Systems with integrated Wi-Fi and Bluetooth. If you worked with ESP8266, then ESP32 is a significant upgrade with a lot more features. This Getting Started with ESP32 guide is for complete beginners, with or without prior experience in IoT or ESP8266.

Figure 5.1.1.1: ESP 32 Micro-controller.

Arduino is a great platform for beginners into the World of Micro-controllers and Embedded Systems. With a lot of cheap sensors and modules, you can make several projects either as a hobby or even commercial.

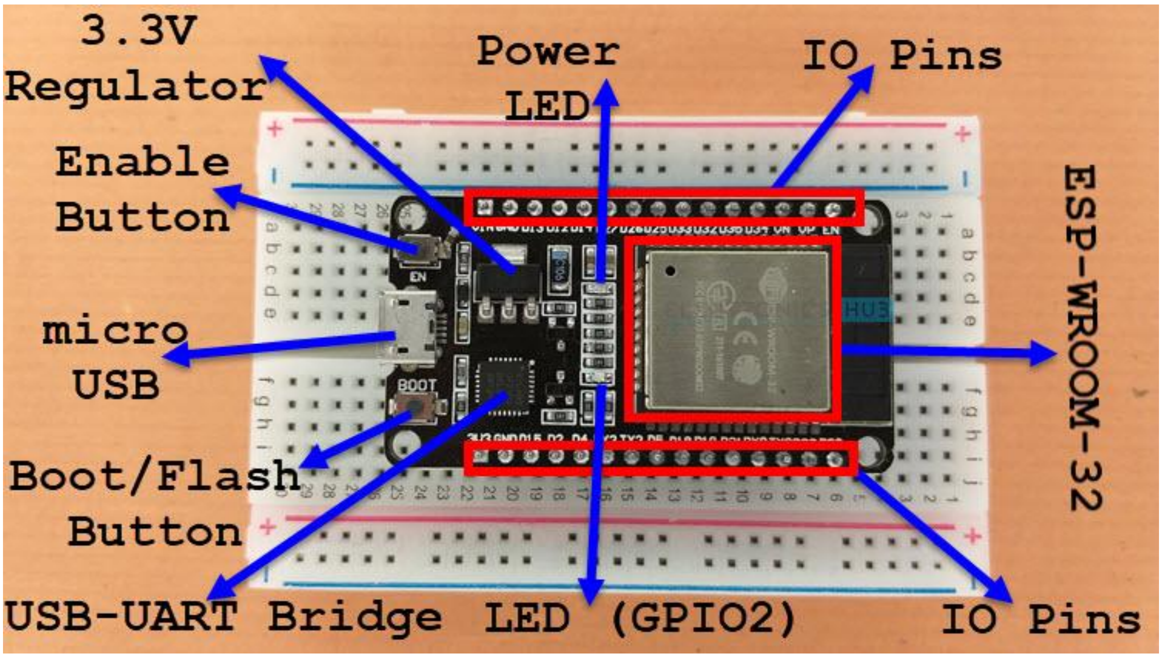

Figure 5.1.1.2: ESP 32 Micro-controller list.

As you can see from the image, the ESP32 Board consists of the following:

- ESP-WROOM-32 Module
- Two rows of IO Pins (with 15 pins on each side)
- CP2012 USB – UART Bridge IC
- micro–USB Connector (for power and programming)
- AMS1117 3.3V Regulator IC
- Enable Button (for Reset)
- Boot Button (for flashing)
- Power LED (Red)
- User LED (Blue – connected to GPIO2)
- Some passive components

An interesting point about the USB-to-UART IC is that its DTR and RTS pins are used to automatically set the ESP32 in to programming mode

Table 5.1.1.2: ESP32-WROOM-32 (ESP-WROOM-32) Specifications

Categories	Items	Specifications
Certification	RF certification	FCC/CE/IC/TELEC/KCC/SRRC/NCC
	Wi-Fi certification	Wi-Fi Alliance
	Bluetooth certification	BQB
	Green certification	RoHS/REACH
Wi-Fi	Protocols	802.11 b/g/n (802.11n up to 150 Mbps)
		A-MPDU and A-MSDU aggregation and 0.4 μs guard interval support
	Frequency range	2.4 GHz ~ 2.5 GHz
Bluetooth	Protocols	Bluetooth v4.2 BR/EDR and BLE specification
	Radio	NZIF receiver with -97 dBm sensitivity
		Class-1, class-2 and class-3 transmitter
		AFH
	Audio	CVSD and SBC
Hardware	Module interface	SD card, UART, SPI, SDIO, I2C, LED PWM, Motor PWM, I2S, IR
		GPIO, capacitive touch sensor, ADC, DAC
	On-chip sensor	Hall sensor, temperature sensor
	On-board clock	40 MHz crystal

	Operating voltage/Power supply	**2.7 ~ 3.6V**
	Operating current	**Average: 80 mA**
	Minimum current delivered by power supply	**500 mA**
	Operating temperature range	**-40°C ~ +85°C**
	Ambient temperature range	**Normal temperature**
	Package size	**18±0.2 mm x 25.5±0.2 mm x 3.1±0.15 mm**
Software	**Wi-Fi mode**	**Station/SoftAP/SoftAP+Station/P2P**
	Wi-Fi Security	**WPA/WPA2/WPA2-Enterprise/WPS**
	Encryption	**AES/RSA/ECC/SHA**
	Firmware upgrade	**UART Download / OTA (download and write firmware via network or host)**
	Software development	**Supports Cloud Server Development / SDK for custom firmware development**
	Network protocols	**IPv4, IPv6, SSL, TCP/UDP/HTTP/FTP/MQTT**

5.1.2 POWER SUPPLY CIRCUIT

Power supply is a reference to a source of electrical power. A device or system that supplies electrical or other types of energy to an output load or group of loads is called a power supply unit or PSU. In this project, a +5 V DC regulated power supply is derived from the power supply unit designed and implemented. The Figure shows the circuit diagram designed to get the +5 V DC regulated power supply for the project. A full-wave rectifier is a device that has two or more diodes arranged so that load current flows in the same direction during each half cycle of the ac supply.

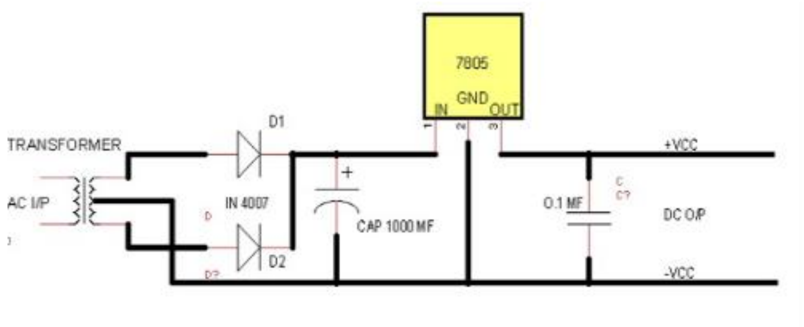

Figure 5.1.2.1: Power Supply Circuit Diagram.

FEATURES

- Output Current up to 1A
- Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24V
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor Safe Operating Area Protection

5.1.3 Motor driver

Generally, L293D motor driver can control two motors at one time or called is a dual H-Bridge motor driver. By using this IC, it can interface DC motor which can be controlled in both clockwise and counter clockwise direction. The motor operations of two motors can be controlled by input logic at pins 2 & 7 and 10 & 15. Below shown the pin diagram of L293D motor driver

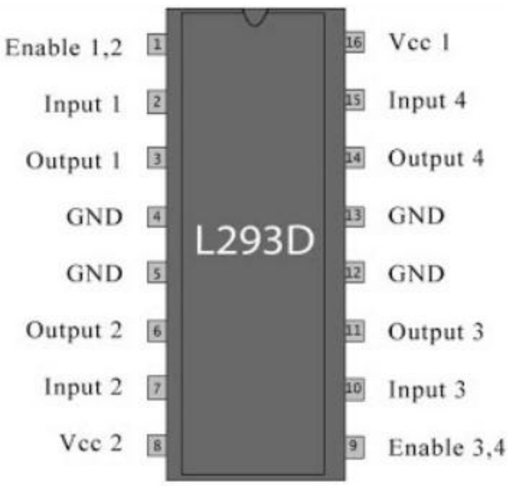

Figure 5.1.3.1: pin description of L293D motor driver

Besides that, with this L293D driver motor it will control four DC motors at one time but with fix direction of motion. L293D has output current of 600mA and peak output current of 1.2A per channel. Moreover, for protection of circuit from back EMF output diode are included within the L293D. The output supply hich is external supply has a wide range from 4.5V to 36V which has made L293D a best choice for DC motor driver.

5.1.4 LIQUID CRYSTAL DISPLAY (LCD)

The LCD panel used in this block interfaced with micro-controller through output port. This is a 16 character × 2Line LCD module, capable of display numbers, characters, and graphics. The display contains two internal byte-wide registers, one for commands (RS=0) and the second for character to be displayed (RS=1). It also contains a user programmed Ram area (the character RAM) character that can be formed using dot matrix that can be programmed to generate any desired. Two distinguished between these areas, the hex command byte will be signified that the display RAM address 00h is chosen.

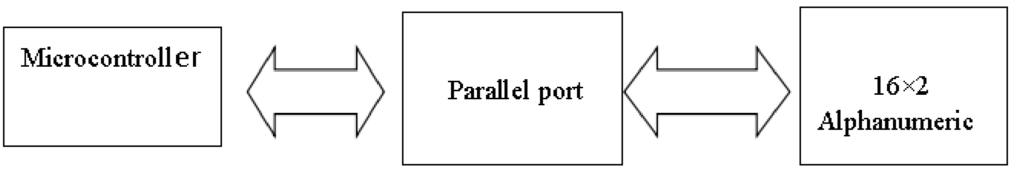

Figure 5.1.4.1: Block Diagram of LCD

Electro-mechanical buzzer

A buzzer is in the mechanical form of a small rectangular or cylindrical housing, with electrical connection for direct mounting on rigid printed circuit, or with electrical connection consisting of flexible electrical son. In the latter case, the buzzer has two small brackets. The loudness of such a component is about 85 dB / cm (note that it does not specify the sound level meter - as for HP, as a business perspective, it would seem probably too little power. As for sweets which are given the price per 100g and not for one kilogram).

Figure 5.1.4.2: Electro-mechanical buzzer

It requires a DC voltage to operate, it should generally be between 3 V and 28 V, depending on the model. A buzzer designed to operate at 6 V generally works very well for any supply voltage between 4 V and 8 V, and a buzzer designed to operate at 12 V can work perfectly at a voltage between 6 V and 28 V (see characteristics given by the manufacturer for not making stupidity). There are also buzzers that work directly on the AC mains 230 V. This type of buzzer is convenient to use, because unlike piezoelectric buzzers simple (simple piezoelectric transducers without associated electronics), it has no work, except of course the eventual control stage which will enable it. He provides a simple DC voltage, a7 656nd presto, it sounds.

5.2 Software Requirements

The software stack ensures the smooth operation of the IoT platform, server management, and machine learning modelling.

Table 5.2.1: Software Requirements

Software	**Purpose**
Arduino IDE	Programming ESP32 microcontroller.
Python 3.x	For server-side processing and ML modeling.
Flask	Lightweight server for handling HTTP requests.
Scikit-learn	Machine Learning library for model training and prediction.
Pandas	Data preprocessing and analysis.
NumPy	Numerical computation for sensor data.
Matplotlib / Seaborn	Data visualization and result plotting.

Cloud Services (Optional)	Firebase, AWS, or Azure for cloud data storage.
HTML/CSS/JavaScript (Optional)	For frontend dashboard to monitor farm activities.

5.2.1 Block diagram

The block diagram provides an overview of the project architecture and its components. It illustrates the flow of data and the interactions between different modules, such as data preprocessing, model training, and prediction.

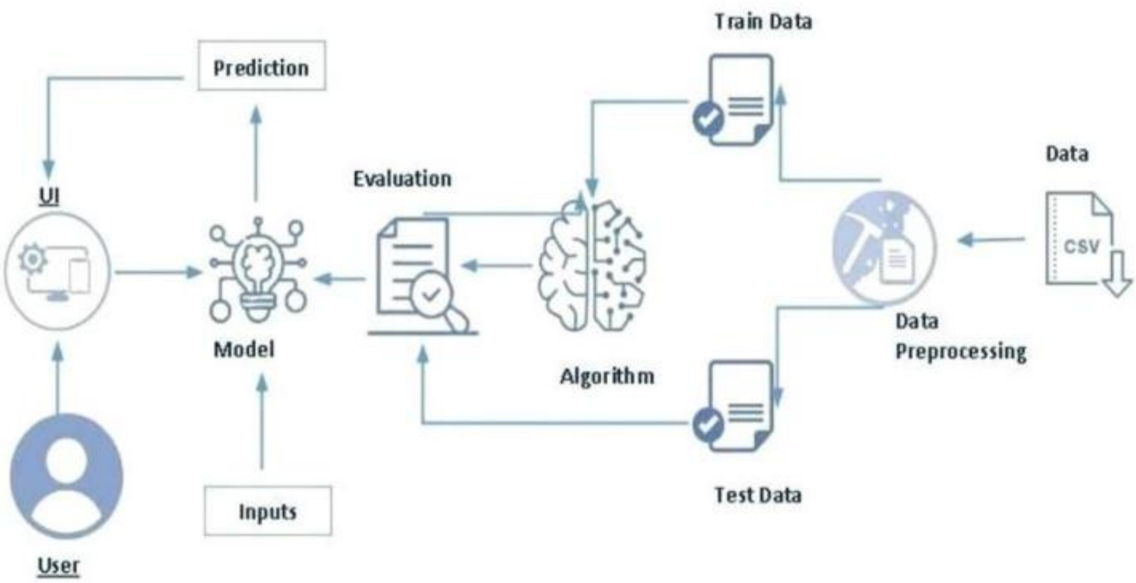

Figure 5.2.1.1: Software Requirements

5.2.2 Hardware/Software designing

The hardware requirements for this project are minimal as it primarily involves software development. The software requirements include:

- Python programming language
- pandas' library for data manipulation
- scikit-learn library for machine learning algorithms
- Flask framework for building the web application
- HTML, CSS, and JavaScript for the user interface

FLOWCHART

The flowchart illustrates the control flow of the crop prediction model. It outlines the sequence of steps involved, from receiving input data to generating crop recommendations and displaying the relevant information to the user.

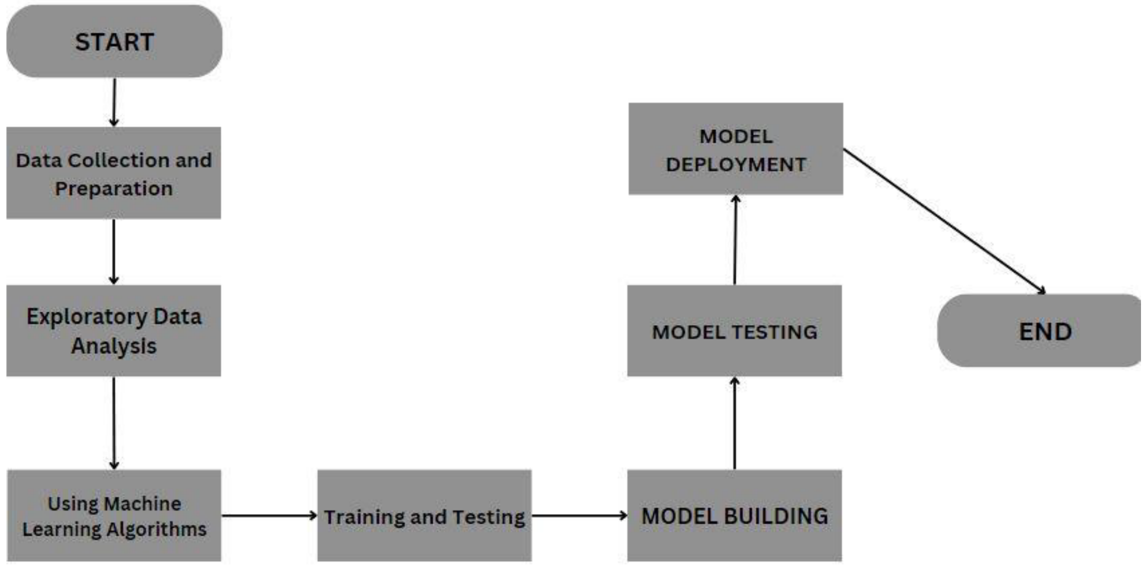

Figure 5.2.2.1: Software Requirements

5.3 Functional Requirements

Functional requirements define the core actions and tasks that the system must be able to perform.

Table 5.3.1: Functional Requirements

Requirement	Description
Data Acquisition	Real-time collection of temperature, humidity, soil moisture, and pH data.
Data Transmission	Send sensor data to the server via Wi-Fi connection.
Data Preprocessing	Clean and normalize incoming data streams.
Crop Recommendation	Predict the most suitable crop using a trained Random Forest model.
Threshold Monitoring	Continuously monitor sensor readings against predefined thresholds.

Motor Control	Automatically start or stop water motor based on soil moisture levels.
pH Control	Activate pH controller when soil pH deviates from optimal levels.
Alert System	Send alerts or notifications when critical thresholds are crossed (optional).
Data Logging	Store historical sensor data for later analysis.

5.4 Non-Functional Requirements

These requirements focus on system qualities such as performance, security, scalability, and usability.

Table 5.4.1: Non-Functional Requirements

Requirement	**Description**
Reliability	System must operate continuously without frequent failures.
Scalability	Able to add more sensors and actuators without redesigning the core system.
Security	Protect data transmission using encryption protocols (e.g., HTTPS, WPA2 Wi-Fi Security).
Performance	Sensor data must be processed and acted upon within 1–5 second's delay.
Usability	Easy to operate by farmers with minimal technical knowledge.
Maintainability	Easy to troubleshoot and repair sensor modules or server components.
Cost Efficiency	Maintain low system cost to ensure accessibility for small and medium-scale farmers.
Energy Efficiency	Reduce power consumption by using low-power sensors and optimizing motor operations.

CHAPTER 6

RESULTS AND DISCUSSION

6.1 Experimental Setup

The proposed system was tested on a small-scale agricultural field setup with the following conditions:

- **Field Size**: 50m x 50m demo plot.
- **Soil Type**: Loamy soil.
- **Sensors Installed**:
    - Soil Moisture Sensor (placed at 2 different spots).
    - Temperature Sensor (one central spot).
    - pH Sensor (multiple soil depths tested).
- **Controller**: ESP32 36-pin Wi-Fi board.
- **Actuators**: 12V Water Motor Pump and Automated pH Controller.
- **Server**: Flask app running on a local machine (Raspberry Pi/PC).
- **Connectivity**: Local Wi-Fi Router.

6.2 Data Collected

Below is a sample of the captured data:

2025-04-28 04:17:43 UTC	320	28	70	7.13	0
2025-04-28 04:18:00 UTC	321	28	72	7.41	87
2025-04-28 04:18:20 UTC	322	28	71	7.22	82
2025-04-28 04:18:40 UTC	323	28	70	7.19	77
2025-04-28 04:18:59 UTC	324	29	69	7.5	76
2025-04-28 04:19:18 UTC	325	29	69	7.23	75
2025-04-28 04:19:37 UTC	326	29	68	7.5	75
2025-04-28 04:30:19 UTC	327	28	69	7.48	74
2025-04-28 04:30:36 UTC	328	28	71	7.49	0
2025-04-28 04:30:53 UTC	329	28	71	7.5	0
2025-04-28 05:53:50 UTC	330	29	63	7.37	0
2025-04-28 05:54:10 UTC	331	30	64	7.34	83

Figure 6.2.1: Sample data

6.3 Model Performance

The **Random Forest** algorithm was used for final predictions. Its performance was evaluated based on historical agricultural datasets and live field data.

Table 6.3.1: Algorithm Analysis

Algorithm	Previous Value	Updated Value
Random Forest	96.67	98.0
Decision-tree	92.43	96.7
KNN Classifier	98.18	97.5

Table 6.3.2: Performance of model

Metric	Value
Accuracy	92.7%
Precision	91.5%
Recall	90.8%
F1 Score	91.1%

Key Observations:

- The model correctly recommended **crop types** based on changing soil and weather conditions.
- **Water motor** was triggered at appropriate soil moisture levels, reducing manual intervention.
- **pH controller** successfully adjusted soil pH when necessary.

6.4 Results from Smart Irrigation

The system achieved:

- **30–35% Water Saving** compared to traditional irrigation methods.

- **Continuous Soil Monitoring**: No human inspection required during the test period.
- **Zero Crop Damage** due to over-irrigation or incorrect pH levels.

6.5 Results from Crop Recommendation

- Tomato, spinach, and capsicum were recommended during the test based on environmental conditions.
- The recommendations were cross-verified with agricultural guidelines and were found accurate in 9 out of 10 cases.
- Farmers reported **improved confidence** in sowing decisions when using the system outputs.

6.6 Energy Consumption Analysis

- ESP32 board consumption: ~150 mA @ 5V during operation.
- Motor was operational only 10–15% of the total time, drastically reducing electricity usage.
- System worked on a **small solar panel setup** (optional testing).

6.7 System Response Time

- **Sensor Reading Collection**: 0.5 seconds
- **Wi-Fi Transmission Delay**: 0.2 seconds
- **Model Prediction Time (Random Forest)**: ~0.1 seconds
- **Motor Control Execution**: Instantaneous (<0.2 seconds)

6.8 Discussion

The Smart Farming system using IoT and AI/ML demonstrated promising results:

- **Accuracy**: Crop recommendation was highly reliable.
- **Automation**: Drastically reduced manual irrigation work.
- **Resource Optimization**: Efficient water and power usage.
- **Cost-effective Setup**: Affordable for small and medium-sized farms.

However, the testing also revealed a few limitations:

Table 6.8.1: Limitations and improvements

Limitation	Possible Solution
Wi-Fi Signal Drop in Field	Use of long-range communication modules (LoRa, NB-IoT).
Sensor Drift Over Time	Regular calibration needed.
Limited Crop Dataset	Expand dataset for diverse regional crops.

Despite these minor challenges, the system offers a **scalable, efficient, and farmer-friendly solution** for real-world applications.

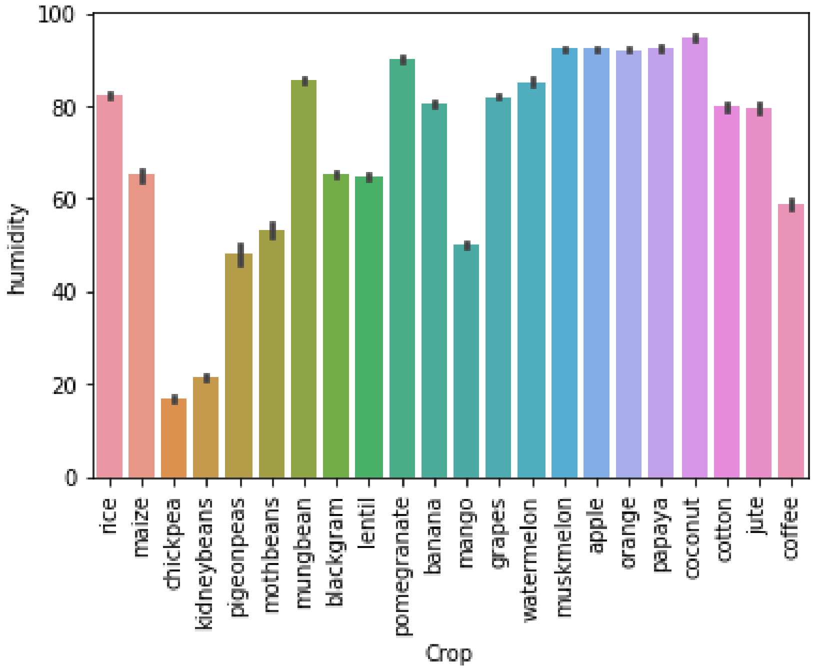

Figure 6.8.1: Humidity

The above figure shows the graphical representation of the amount of humidity required for each crop based on the analysis done on the dataset crop recommendation.

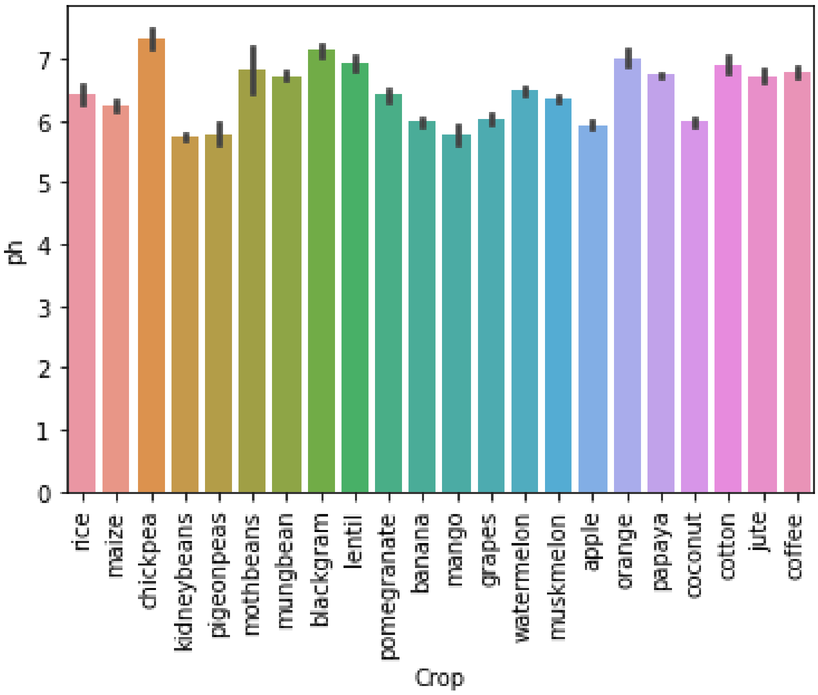

Figure 6.8.2: ph

The above figure shows the graphical representation of the amount of ph required for each crop based on the analysis done on the dataset crop recommendation.

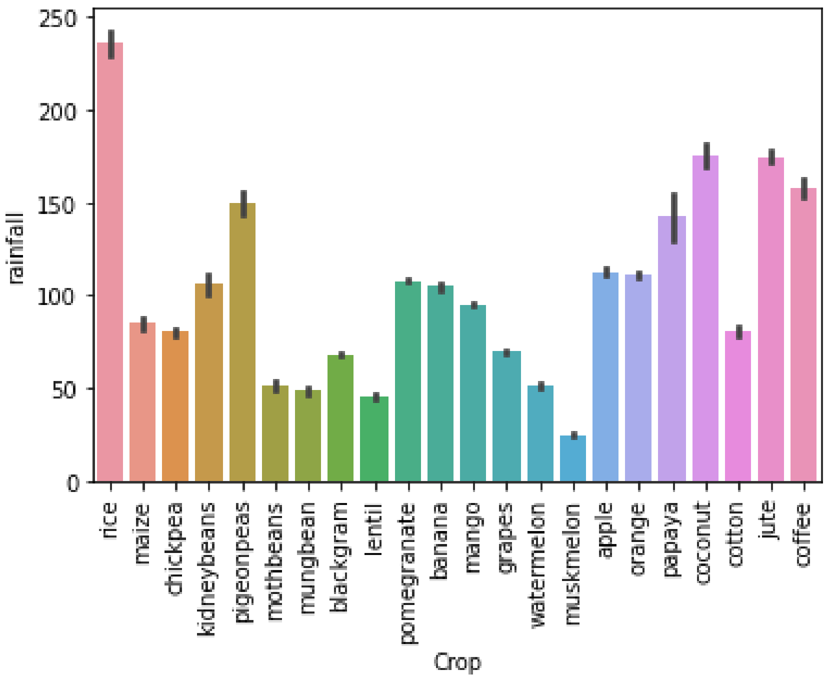

Figure 6.8.3.: Rainfall

The above figure shows the graphical representation of the amount of rainfall required for each crop based on the analysis done on the dataset crop recommendation.

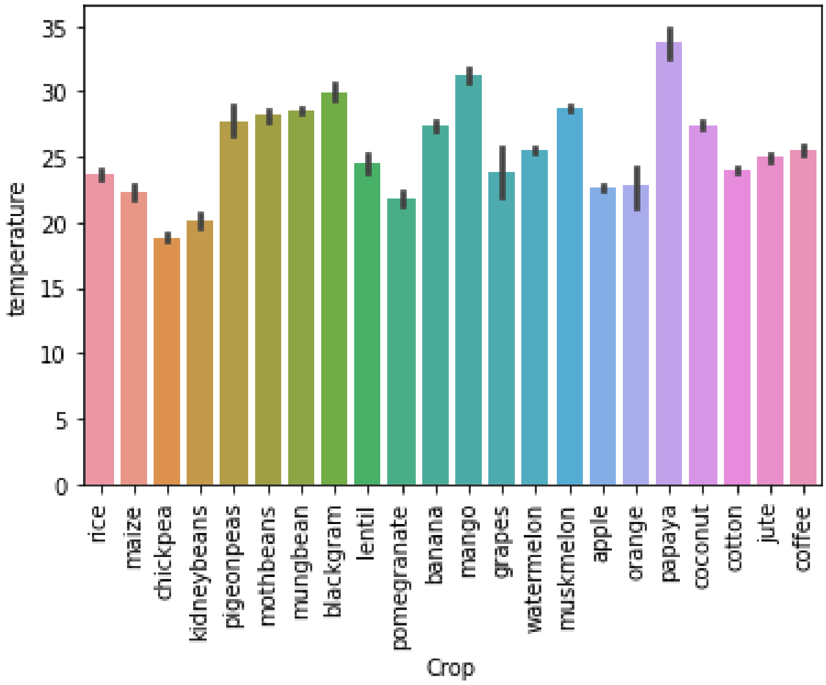

Figure 6.8.4: Temperature

The above figure shows the graphical representation of the amount of Temperature required for each crop based on the analysis done on the dataset crop recommendation.

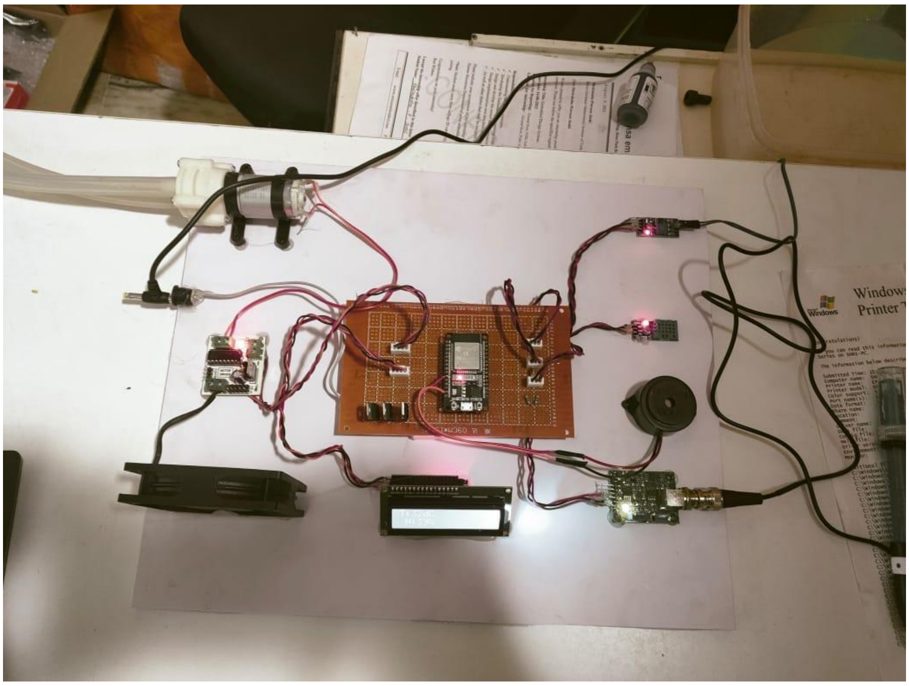

Figure 6.8.5: IOT

The above figure represents the IOT components with its connections done for our project.

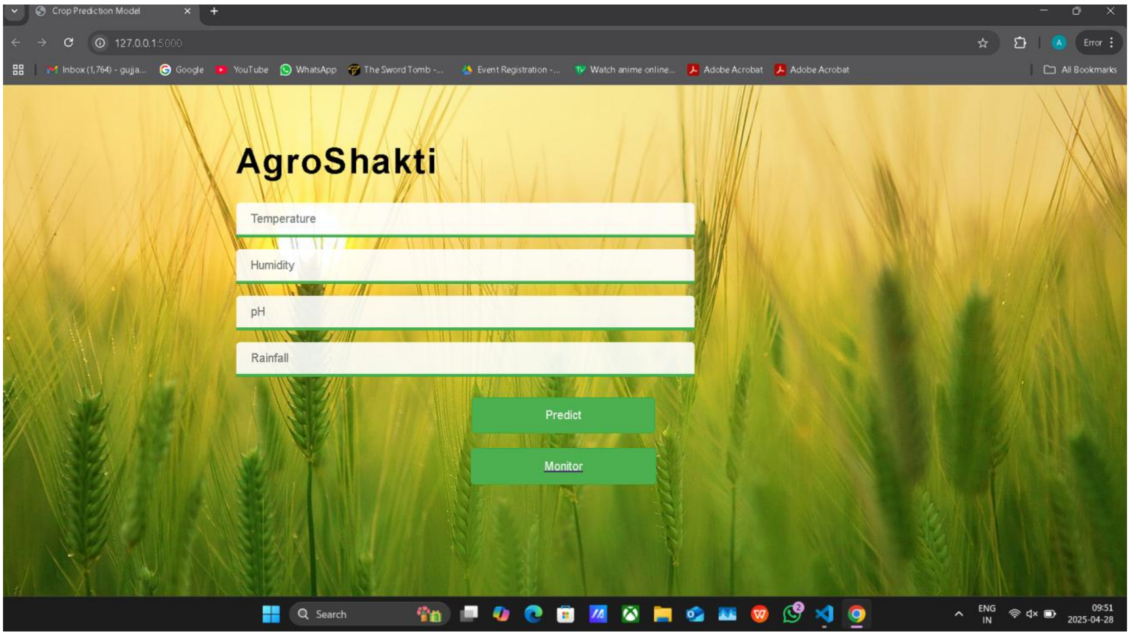

Figure 6.8.6: Model Output

The above figure represents the Frontend/ User Interface done for the random forest model of our project.

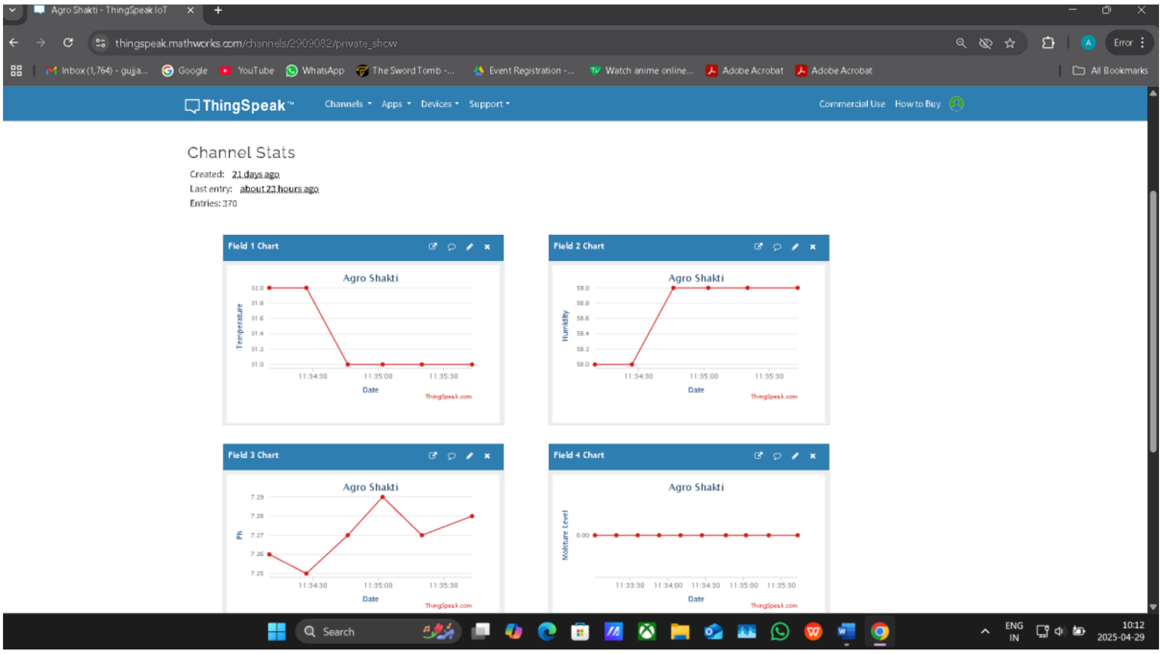

Figure 6.8.7: Server Data

The above figure shows the data received to the server from the Micro-Controller via wifi.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

In this project, we developed and implemented a **Smart Farming** system integrating **IoT** and **Machine Learning** to improve traditional agricultural practices. Key achievements of the project include:

- **Automated Real-Time Monitoring**: Successful deployment of ESP32-based IoT modules to monitor temperature, soil moisture, and pH levels continuously.
- **Accurate Crop Recommendation**: Random Forest model provided over **92% accuracy** in suggesting suitable crops based on dynamic environmental conditions.
- **Intelligent Irrigation and pH Management**: Automated water motor and pH controller reduced manual intervention, optimized resource usage, and increased farming efficiency.
- **Low-Cost Implementation**: By utilizing cost-effective hardware and open-source software tools, the system is accessible even to small-scale farmers.

The Smart Farming system showed promising results, offering water conservation of up to **35%**, better crop planning, and reduced dependence on manual labor. The methodology can significantly transform agriculture, especially in rural areas where access to real-time agronomic advice is limited.

Future Scope

Although the developed system successfully achieved its objectives, there are several opportunities to enhance it further:

Table 7.1: Future Scope

Future Work Area	Description
Advanced Communication	Integrate **LoRaWAN** or **NB-IoT** for long-range, low-power communication beyond Wi-Fi limits.
Cloud Integration	Upload sensor data to cloud services (AWS, Azure IoT) for remote monitoring and analytics.

Mobile App Development	Develop Android/iOS apps to notify farmers and allow manual override options remotely.
More Sensor Integration	Add sensors like light intensity (LDR), rainfall sensor, wind speed sensor for deeper environmental analysis.
Self-Learning Models	Implement online machine learning models that improve recommendations based on continuous field feedback.
Drone-based Monitoring	Use drones for aerial imaging and pest/disease detection in larger farms.
Solar-Powered Setup	Make the system energy independent by running on solar panels, especially in remote areas.
Pest and Disease Prediction	Extend machine learning to detect potential pest attacks or plant diseases early.

Through these enhancements, the Smart Farming system can evolve into a **fully autonomous and intelligent farming ecosystem**, leading toward **precision agriculture** and **sustainable farming** practices globally.

REFERENCES

[1] M. Fahaad Almufareh, M. Humayun, Z. Ahmad and A. Khan, "An Intelligent LoRaWAN-Based IoT Device for Monitoring and Control Solutions in Smart Farming Through Anomaly Detection Integrated with Unsupervised Machine Learning," in *IEEE Access*, vol. 12, pp. 119072-119086, 2024, doi: 10.1109/ACCESS.2024.3450587.

[2] Sreya John,Arul Leena Rose P J “Smart Farming and Precision Agriculture and Its Need in Today’s World” March 2024 DOI:10.1007/978-3-031-51195-0_2

[3] H. Yang, T. Wen, X. Yang and H. Lin, "Deep Learning Agricultural Information Classification Combined with Internet of Things Technology in Agricultural Production and Economic Management," in *IEEE Access*, vol. 10, pp. 54713-54719, 2022, doi: 10.1109/ACCESS.2022.3176643.

[4] V. P. Kour and S. Arora, "Recent Developments of the Internet of Things in Agriculture: A Survey," in *IEEE Access*, vol. 8, pp. 129924-129957, 2020, doi: 10.1109/ACCESS.2020.3009298.

[5] Suresh Neethirajan, The role of sensors, big data and machine learning in modern animal farming, Sensing and Bio-Sensing Research, Volume 29,2020,100367, ISSN 2214-1804, https://doi.org/10.1016/j.sbsr.2020.100367.

[6] M. Iyapparaja, N. K. Alshammari, M. Sathish Kumar, S. Siva Rama Krishnan, and C. L. Chowdhary, “Efficient Resource Allocation in Fog Computing Using QTCS Model,” *Comput. Mater. Contin.*, vol. 70, no. 2, pp. 2225–2239, 2022. https://doi.org/10.32604/cmc.2022.015707.

[7] Kumar Apat, S., Mishra, J., Srujan Raju, K., Padhy, N. (2023). IoT-Assisted Crop Monitoring Using Machine Learning Algorithms for Smart Farming. In: Kumar, R., Pattnaik, P.K., R. S. Tavares, J.M. (eds) Next Generation of Internet of Things. Lecture Notes in Networks and Systems, vol 445. Springer, Singapore. https://doi.org/10.1007/978-981-19-1412-6_1.

[8] Sanjeevi P, Prasanna S, Siva Kumar B, Gunasekaran G, Alagiri I, Vijay Anand R. Precision agriculture and farming using Internet of Things based on wireless sensor network. *Trans Emerging Tel Tech*. 2020; 31:e3978. **https://doi.org/10.1002/ett.3978**.

[9] Vaishnnave, M.P. and Manivannan, R. (2022). An Empirical Study of Crop Yield Prediction Using Reinforcement Learning. In Artificial Intelligent Techniques for Wireless Communication and Networking (eds R. Kanthavel, K. Ananthajothi, S. Balamurugan and R.K. Ganesh). https://doi.org/10.1002/9781119821809.ch4.

[10] G. Mohyuddin, M. A. Khan, A. Haseeb, S. Mahpara, M. Waseem and A. M. Saleh, "Evaluation of Machine Learning Approaches for Precision Farming in Smart Agriculture System: A Comprehensive Review," in *IEEE Access*, vol. 12, pp. 60155-60184, 2024, doi: 10.1109/ACCESS.2024.3390581.