HyGrow- A smart IoT-Driven Plant Watering and Environmental Monitoring System Integrating Real-Time Temperature, Humidity, and Soil Moisture Sensing for Optimal Plant Care Management

A PROJECT REPORT

Submitted by

ARPUDHA SOUNDARARAJAN (2022115068)

JAGADHEESWARI.V (2022115069)

PREETHI. B (2022115305)

In partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION SCIENCE AND TECHNOLOGY

DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY

COLLEGE OF ENGINEERING GUINDY

ANNA UNIVERSITY,

CHENNAI 600025

NOVEMBER 2024

ANNA UNIVERSITY

CHENNAI-600025

BONAFIDE CERTIFICATE

Certified that this project report titled **"HyGrow - A smart IoT-Driven Plant Watering and Environmental Monitoring System Integrating Real-Time Temperature, Humidity, and Soil Moisture Sensing for Optimal Plant Care Management"** is a bonafide work of ARPUDHA SOUNDARARAJAN (2022115068), JAGADHEESWARI V (2022115069) and PREETHI B (2022115305) under my supervision for SUMMER PROJECT (IT5513). Certified further that to the best of my knowledge and belief, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion on this or any other candidate.

PLACE: CHENNAI
DATE:

Dr. K. ARULDEEPA
PROFESSOR
PROJECT GUIDE
DEPARTMENT OF IST, CEG
ANNA UNIVERSITY
CHENNAI-600025

COUNTERSIGNED

Dr. S. SWAMYNATHAN
HEAD OF THE DEPARTMENT
DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY
COLLEGE OF ENGINEERING GUINDY
ANNA UNIVERSITY
CHENNAI-600025

ABSTRACT

HyGrow represents a significant advancement in agricultural technology, offering an innovative solution to the pressing issues of water scarcity and resource management in farming. This smart irrigation system integrates Internet of Things (IoT) technology with precision agriculture principles to revolutionize water management practices. At its core, HyGrow utilizes sophisticated sensors connected to an ESP32 microcontroller, continuously monitoring critical environmental parameters such as soil moisture, ambient temperature, and humidity. The system's strength lies in its real-time data collection and analysis capabilities. By leveraging machine learning algorithms, particularly Linear Regression, HyGrow processes the gathered environmental data to make informed decisions about irrigation needs. This predictive approach enables the system to optimize water usage, ensuring that plants receive the required water, thus minimizing waste and promoting healthier crop growth.

HyGrow's user interface, comprising a web application and a mobile app, offers unprecedented control and monitoring capabilities to farmers and agricultural professionals. The implementation of HyGrow has shown promising results in initial trials, demonstrating significant improvements in water efficiency and crop health. By providing a data-driven approach to irrigation, the system addresses the dual challenges of resource conservation and agricultural productivity. As global concerns about water scarcity and sustainable farming practices continue to grow, HyGrow emerges as a timely and practical solution, paving the way for more sustainable and technologically advanced agricultural methods.

ACKNOWLEDGEMENT

It is our privilege to express our deepest sense of gratitude and sincere thanks to **Dr. K. ARULDEEPA** Professor, Project Guide, Department of Information Science and Technology, College of Engineering, Guindy, Anna University, for her constant supervision, encouragement, and support in our project work. We greatly appreciate the constructive advice and motivation that was given to help us advance our project in the right direction.

We are grateful to **Dr. S. SWAMYNATHAN**, Professor and Head, Department of Information Science and Technology, College of Engineering Guindy, Anna University for providing us with the opportunity and necessary resources to do this project.

We would also wish to express our deepest sense of gratitude to the Members of the Project Review Committee: **Dr. M. VIJAYALAKSHMI** and **Mr. H. RIASUDHEEN** Professors, Department of Information Science and Technology, College of Engineering Guindy, Anna University, for their guidance and useful suggestions that were beneficial in helping us improve our project.

We also thank the faculty members and non-teaching staff members of the Department of Information Science and Technology, Anna University, Chennai for their valuable support throughout the course of our project work.

ARPUDHA SOUNDARARAJAN (2022115068)
JAGADHEESWARI.V (2022115069)
PREETHI. B (2022115305)

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Agriculture has been the bedrock of human civilization, playing a crucial role in economic stability and growth throughout history. However, traditional farming practices face unprecedented challenges. Water scarcity, inefficient resource utilization, and labour-intensive processes are becoming increasingly problematic, threatening food security and environmental sustainability on a global scale.The advent of the Internet of Things (IoT) technology has ushered in a new era of possibilities for agriculture.

IoT in agriculture facilitates real-time data collection and improved resource management, enabling farmers to make informed decisions based on accurate, up-to-date information about their crops and environmental conditions. It provides real-time data for users to remotely control the motor, ensuring efficient watering. Smart irrigation systems, powered by IoT, provide precise water control through continuous monitoring of soil conditions, weather patterns, and water usage.

The benefits of such systems extend beyond mere water conservation; they also contribute to reducing labor costs and increasing crop yields. Water inefficiency remains a significant concern, with many irrigation systems still wasting substantial amounts of water through overwatering or poor distribution. These challenges underscore the need for more sophisticated and tailored solutions that can address the specific needs of modern agriculture.

HyGrow, is designed to tackle these challenges head-on. By developing a smart irrigation system that leverages IoT technology, we aim to provide users with unprecedented control over their irrigation processes. The system enables remote monitoring and management of water distribution to crops, ensuring optimal water usage while significantly reducing the need for manual intervention. This approach not only promotes better water conservation but also contributes to overall soil health by preventing issues related to over or under-watering. While IoT has already made significant strides in advancing agricultural practices, HyGrow takes these improvements a step further. Our system offers a practical, user-friendly tool for effective irrigation management by integrating real-time data collection, machine learning analysis, and remote-control capabilities, HyGrow represents a comprehensive solution for optimizing resource use and enhancing agricultural efficiency. The potential impact of systems like HyGrow extends beyond individual farms.

By optimizing water usage, IoT-based irrigation systems contribute significantly to the sustainable use of natural resources. This not only helps in preserving ecosystems but also reduces the environmental footprint of farming activities. As global water resources come under increasing pressure due to climate change and population growth, the role of smart irrigation in sustainable agriculture becomes ever more critical. Moreover, the implementation of IoT in irrigation systems is revolutionizing the agricultural sector by making irrigation more efficient, sustainable, and cost-effective.

This transformation is crucial for meeting the growing global demand for food in a world where resources are increasingly constrained. By enabling more precise and data-driven farming practices, systems like HyGrow are at the forefront of efforts to enhance food security while minimizing environmental impact.

CHAPTER 2

LITERATURE SURVEY

2.1 Research

Smart irrigation systems have gained significant attention in recent years due to the increasing need for efficient water management in agriculture, particularly in regions facing water scarcity. Various studies have explored the integration of Internet of Things (IoT) technologies and sensor networks to enhance irrigation practices. These technologies help optimize water usage, reduce human intervention, and improve agricultural productivity, while also contributing to sustainable development goals by conserving resources and addressing climate challenges. The journey of smart irrigation systems began with the recognition of water scarcity as a critical issue in agriculture. Chavan (Chavan 2023) set the stage by implementing a smart drip irrigation system using soil moisture sensors. This pioneering work demonstrated the potential of automation in water conservation, showing how simple sensor-based systems could significantly reduce water waste.

Chavan's research highlighted the need for more sophisticated solutions that could adapt to varying environmental conditions, laying the groundwork for future innovations like HyGrow. (Kanwal 2023) Kanwal et al. explored the use of machine learning models to predict wheat yield variations in semi-arid regions. By analyzing historical yield data alongside climate variables, the study demonstrated that models like random forests effectively forecast yield variability. This research highlights the potential of machine learning in addressing climate-induced challenges in agriculture. (Sajjad et al.) Sajjad et al. reviewed the effects of plant growth regulators (PGRs) on ornamental plants. The review covered various PGRs such as auxins, gibberellins, and cytokinins, detailing their role in promoting growth, flowering, and root development. The study offered valuable insights into how PGRs can be used to enhance ornamental plant production and improve aesthetic qualities, Building on this foundation, El Mezouari et al. (ElMezouari2022) expanded the concept by integrating multiple sensors into a comprehensive framework. Their research emphasized the importance of system architecture in creating efficient irrigation systems, particularly in arid regions.

By combining data from various sensors, El Mezouari's team showed how a more holistic approach to environmental monitoring could lead to better irrigation decisions. This multi-sensor approach directly influenced the design of HyGrow, which incorporates not only soil moisture but also temperature and humidity sensors for a more complete understanding of plant needs.As the field progressed, García et al. (Garcia2020) provided a crucial overview of IoT trends in smart irrigation. Their work underscored the growing importance of data analytics and machine learning in optimizing irrigation schedules.

García's team explored how real-time data collection and analysis could lead to more precise water management, reducing waste while improving crop health. HyGrow takes this concept further by implementing a Linear Regression model to predict humidity based on temperature, allowing for more accurate and anticipatory watering decisions. This predictive capability sets HyGrow apart from reactive systems, enabling proactive management of water resources.Khaled et al. (Khaled2022)delved deeper into the technological components of IoT-based irrigation, discussing the integration of hardware and software. Their research highlighted the need for customizable solutions that could adapt to diverse environments. Khaled's work emphasized the importance of scalability and flexibility in smart irrigation systems, recognizing that different crops and regions require tailored approaches. HyGrow addresses this challenge by offering a flexible system that can be calibrated for different crop types and local conditions, making it suitable for a wide range of agricultural applications.

The advent of cloud computing and mobile applications in smart irrigation, as explored by Kumar et al. (Kumar2022), marked a significant leap forward in accessibility and control. Kumar's team demonstrated how cloud-based systems could provide farmers with unprecedented access to their irrigation data and controls, regardless of their physical location. HyGrow capitalizes on this trend with its user-friendly web and mobile interfaces, allowing farmers to monitor and manage their irrigation systems remotely.

This feature not only enhances convenience but also enables rapid response to changing conditions, a crucial factor in optimizing water use and crop health. Masson-Delmotte et al. (MassonDelmotte2020) brought attention to the role of smart irrigation in mitigating the effects of climate change on agriculture. Their comprehensive report highlighted how precision agriculture technologies could help farmers adapt to increasingly unpredictable weather patterns. HyGrow's predictive capabilities and real-time environmental monitoring align perfectly with this goal, offering a solution that can help farmers navigate the challenges posed by climate change. By providing data-driven insights, HyGrow empowers farmers to make informed decisions in the face of environmental uncertainty (Raza Ahmad) Ahmad et al. developed a portable humidity meter for monitoring agricultural product preservation.

The study emphasized the importance of real-time humidity control to prevent spoilage, ensuring longer shelf life for perishable goods. The device’s fast response and affordability make it particularly useful for small-scale farmers. (Razal) Raza et al. examined the impact of irrigation reforms on wheat productivity in Punjab. Their study focused on participatory irrigation management and water use efficiency. Findings revealed that irrigation policy changes significantly improved crop yields, stressing the need for proper water resource management to boost agricultural productivity. (Sarwar et al.) Sarwar et al. analyzed how different irrigation levels affect wheat cultivars. The study showed that proper irrigation not only increases yield but also improves key yield components like grain weight.

This Project suggests that precise irrigation is crucial for maximizing wheat production in water-scarce environments. While Obaideen et al. (Obaideen2021) discussed challenges in implementing IoT solutions in agriculture, including data privacy and security, their work also highlighted the immense potential of these technologies. Obaideen's team emphasized the need for robust, secure systems that could protect farmers' data while providing valuable insights. HyGrow addresses these concerns through its integrated approach, combining local processing with secure cloud connectivity to strike a balance between data accessibility and privacy protection. Said Mohamed et al. (SaidMohamed2021) further emphasized the role of smart farming technologies in improving agricultural management.

Their research showcased how IoT and AI could transform traditional farming practices, leading to more sustainable and productive agriculture. HyGrow builds upon this concept by not only monitoring conditions but also using machine learning to make intelligent irrigation decisions, pushing the boundaries of what's possible in precision agriculture. Early pioneers like Saraf et al. (Saraf2017) laid the groundwork for IoT-based monitoring and control systems in agriculture. Their work on sensor integration and automated control systems demonstrated the feasibility of smart irrigation on a practical level.

HyGrow represents the next evolution of these systems, combining advanced sensors, machine learning, and user-friendly interfaces to create a comprehensive smart irrigation solution that builds on the foundations laid by Saraf and others. Finally, Vallejo-Gómez et al. (VallejoGomez2023) conducted a systematic review of smart irrigation technologies, highlighting the ongoing innovation in the field. Their work provided a comprehensive overview of the state of smart irrigation, identifying key trends and challenges. HyGrow stands out in this landscape by offering a unique combination of features: real-time environmental monitoring, machine learning-based predictions, remote control capabilities, and a user-friendly interface.

2.2 Conclusion from the Literature Survey

HyGrow positions itself at the forefront of smart irrigation technology. In summary, HyGrow builds upon the foundational work of previous researchers, addressing key challenges identified in the literature while incorporating cutting-edge technologies. Its main advantages include:

- Comprehensive environmental monitoring (soil moisture, temperature, and humidity)
- Machine learning-based predictive capabilities for optimized irrigation
- User-friendly remote control through web and mobile interfaces
- Flexibility to adapt to various crop types and environmental conditions.
- Integration of local processing and cloud connectivity for enhanced security and accessibility.

These features position HyGrow as a cutting-edge solution in the evolving landscape of smart irrigation systems, offering farmers a powerful tool to enhance water efficiency, crop health, and overall agricultural productivity.

By learning from and building upon the rich history of smart irrigation research, HyGrow represents a significant step forward in the quest for sustainable, efficient agriculture in the face of global challenges.

CHAPTER 3

PROBLEM STATEMENT

Water scarcity and inefficient irrigation methods are major issues in agriculture resulting in overwatering, under-watering, and resource waste. The traditional irrigation systems lack real-time data and adapting to environmental conditions. These inefficiencies result in detoriation of crop health, higher operational costs, and unsustainable water usage.

Due to scarcity, there is an urgent need to find new solutions to optimize water use while ensuring sustainable agriculture. With the increasing global concerns about water scarcity and permaculture practices, HyGrow comes out in a timely and impactful manner, paving the way for sustainable agriculture and technology.

Objectives

- To develop a smart IoT-based plant watering system.
- To monitor temperature, humidity, and soil moisture in real-time.
- To provide remote control capabilities for efficient watering.
- To enhance water efficiency and plant health using IoT and data analysis.
- To contribute to sustainable agricultural practices.

CHAPTER 4

PROPOSED SYSTEM OF HYGROW

The figure 4.1 explains how the architecture flow is happening in this project starting from, collecting the dataset from the hardware and developing the model using the data.

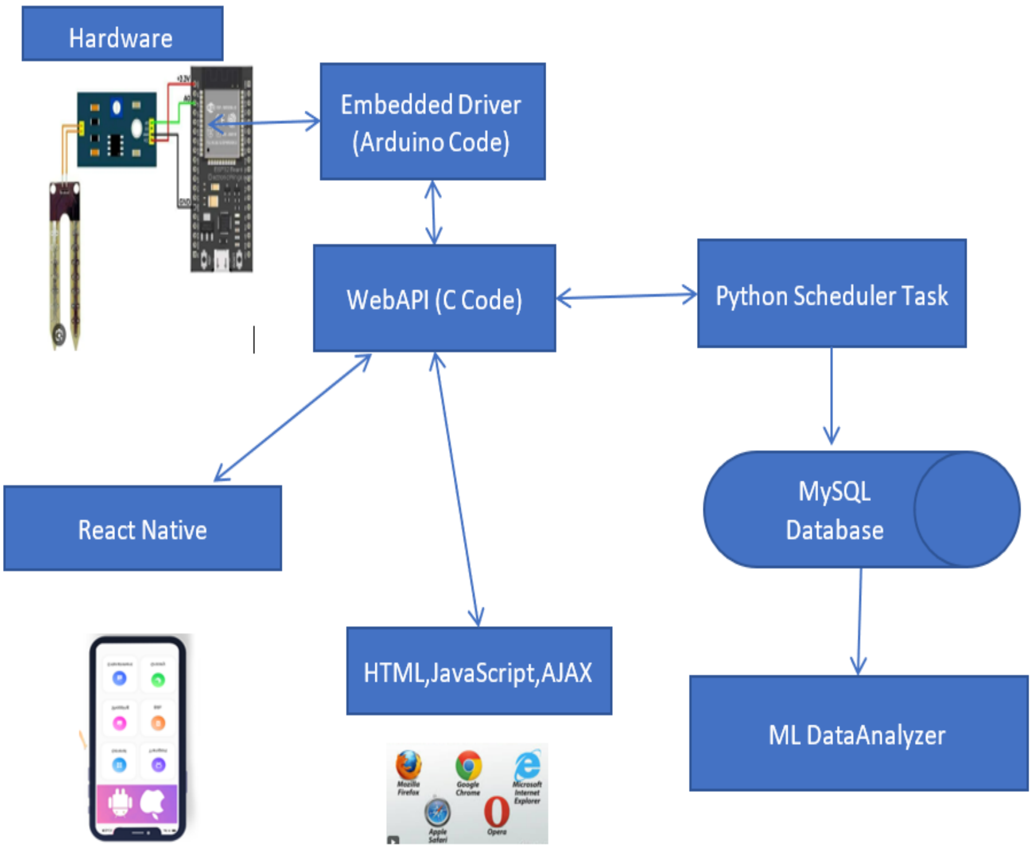

Fig 4.1 System Architecture and Data flow

Fig 4.1 illustrates the architecture and data flow for an IoT-based plant watering system. At its foundation, the system uses soil moisture sensors connected to a microcontroller, such as an ESP32 or Arduino, which continuously measures the moisture levels of the soil. The microcontroller is programmed with Arduino-based embedded code, enabling it to process the sensor readings and communicate them to a centralized WebAPI developed in C.

This WebAPI acts as a hub, facilitating seamless communication between the hardware and other components of the system. A Python scheduler task interacts with the WebAPI at regular intervals to retrieve updated sensor data, which is then stored in a MySQL database for long-term storage and analysis. The database serves as the primary data source for a machine learning module that processes the collected data to generate insights, such as predicting future watering requirements or identifying patterns in soil moisture levels.

Users can interact with the system through two primary interfaces: a mobile application built using React Native and a web interface accessible through browsers. Both platforms communicate with the WebAPI to display real-time data and provide controls for activating the water pump motor, allowing users to remotely manage and monitor the watering process.

By integrating hardware, software, and machine learning, this system ensures efficient water usage, reducing waste and supporting plant health through precise irrigation management.

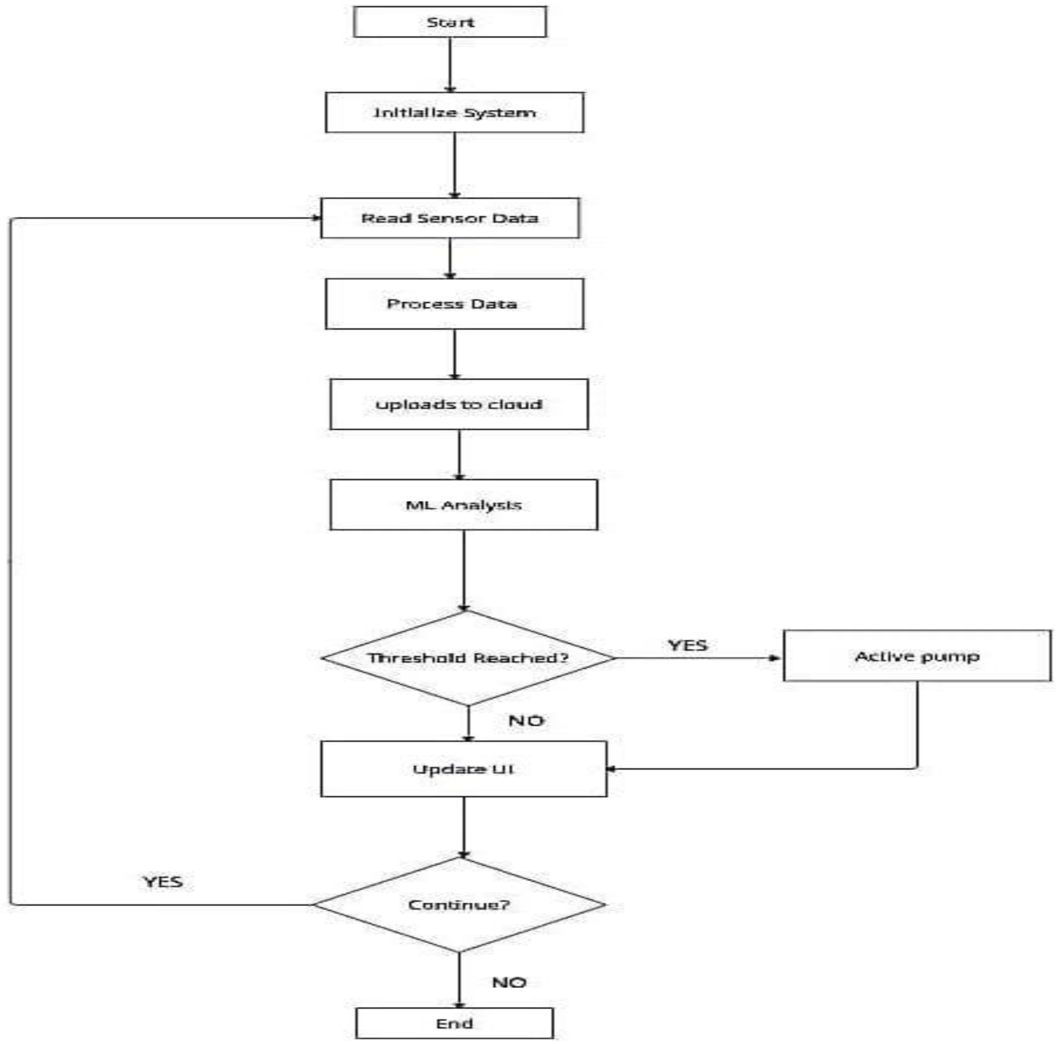

Figure 4.2 Systematic flowchart of HyGrow

Fig 4.2 explains the system flowchart. Upon initialization, the hardware reads data from the sensors and uploads it to the cloud sketch for processing. Leveraging machine learning, the data is analyzed, and graphs are generated using a linear regression algorithm. The motor can be remotely controlled through user interfaces, which also provide real-time temperature and humidity information.

4.1 CIRCUIT DIAGRAM

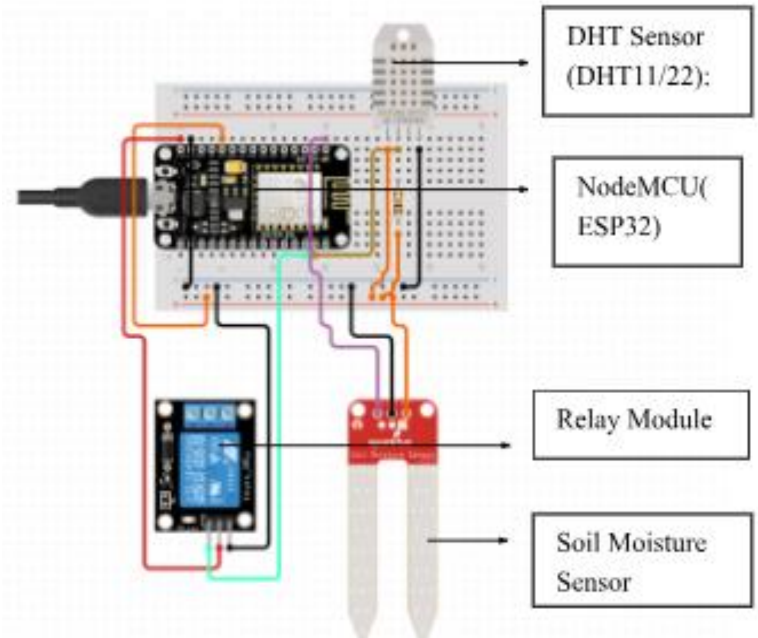

Fig 4.3 Circuit Connection

Fig 4.3 explains how the circuit is connected in the prototype. This circuit is designed to monitor environmental conditions. It facilitates communication between sensors, the control system, and actuators like the water pump, ensuring efficient water usage based on real-time data.

4.2 COMPONENT DESCRIPTION

4.2.1. Hardware Components

NodeMCU ESP32

The ESP32 NodeMCU is a versatile microcontroller with built-in Wi-Fi, making it ideal for IoT applications. It serves as the brain of our system, connecting to sensors, processing data, and managing communication with the cloud server. Key features include:

- Dual-core processor operating at 240MHz
- 520 KB SRAM and 4MB Flash memory
- Integrated Wi-Fi and Bluetooth
- Multiple GPIO pins for sensor and actuator connections
- Data update rate: Every 1-2 seconds

In our system, the ESP32 is programmed using the Arduino IDE, enabling seamless integration with various sensors and control of the irrigation valves based on real-time data and user-defined parameters. This dual-function sensor allows our system to consider both air temperature and humidity when making irrigation decisions, ensuring a more holistic approach to plant care.

Soil Moisture sensor

The soil moisture sensor is crucial for detecting the moisture level in the soil. We employ a capacitive sensor for its durability and ac curacy. Key aspects include:

- Measures soil moisture through dielectric constant changes
- Corrosion-resistant compared to resistive sensors
- Operating voltage: 3.3V to 5V
- Output: Analog signal (0-1023)

This sensor provides essential data for deter mining when irrigation is necessary, helping to prevent both under and over-watering of crops.

DHT sensor (DHT 11/22)

The DHT sensor measures both temperature and humidity, providing crucial environmental data. Specifications include:

- Temperature range: 0-50°C (±2°C accuracy)
- Humidity range: 20-80
- Resolution: 1°C (temperature)

Relay Module (1- channel)

The relay module acts as a switch, control ling high-voltage devices (like water pumps) through low-voltage signals from the micro controller. Features include:

- Operating voltage: 5V DC
- Can control AC appliances up to 250V
- LED indicator for relay status

In our system, the relay module is crucial for activating and deactivating the water pump based on the irrigation decisions made by the control algorithm.

4.2.2. Software

The Arduino code for the ESP32 is designed to interact with various sensors, process the collected data, and control actuators. On the server side, machine learning algorithms, such as linear regression, are utilized to predict water demand based on real-time data inputs.

MySQL is used to store environmental and system information for future reference and analysis. Python code is used to retrieve real-time data and store it in the database.

A mobile app is built using react native
A web application is built using Web API, HTML, CSS and JavaScript

4.2.3 User Interface

Users will be able to monitor real-time data, manually adjust irrigation settings, and receive notifications with the help of a web-based dashboard and mobile app.

4.3 PSEUDO-CODE

Getting data from the hardware using embedded driver
Begin server

```
// Loop function
Loop:
   // Examine Sensors
   Read data from soil moisture sensor
   Examine temperature and humidity from DHT sensor

   If DHT readings are valid:
      Print humidity and temperature to Serial Monitor
      Show humidity and temperature on Liquid Crystal Display

   // Test Soil Moisture and manipulate Relay
   If soil moisture is very dry:
      turn motor on
      Show "Soil Dry" and "Motor ON"
   Else if soil moisture is between dry and moist:
      turn motor off
      display "Motor OFF"
   Else if soil is moist:
      flip motor off
      display "Motor OFF"

   // Ship Data records to Server Periodically
   If 10 seconds have exceeded:
      If WiFi is connected:
         connect to server
         Ship JSON data with temperature and humidity to server
         Print server response to Serial monitor
      Else:
         Print error message
         delay 2 seconds
```

```
Import required libraries:
    logging for logging messages
    Flask for web server framework
    request and jsonify for handling requests and JSON responses
    mysql.connector for MySQL database operations

# Initialize Flask app and configure logging
Create Flask software instance
Set logging stage to info

# connect with MySQL database
connect to MySQL database with use of given credentials
Create a cursor for database operations

# Outline route to receive data
Outline POST route '/mldata' to handle incoming data
    Log "Data Acquired" to indicate data arrival
    Retrieve JSON data from the request
    If data exists:
        Extract temperature and humidity from JSON payload
        Prepare SQL INSERT query to feature temperature and humidity to
        "temperature_humidity" table
        Execute the query with extracted values
        Commit the transaction to keep adjustments
        Log the temperature and humidity values for report-retaining
        return JSON response indicating "success" with HTTP status code 200
    If no records:
        Return JSON response indicating "failure" with HTTP status code 400
# beginFlask app
If script is run directly:
```

Log "Server began" to indicate server start-up

Run the Flask app on specific IP address (192.168.83.41)

Machine Learning

Import necessary libraries:

pandas for managing records

train_test_split and LinearRegression for machine learning

mysql.connector for MySQL database

matplotlib.pyplot for statistics visualization

MySQL connection and information retrieval

connect with MySQL database with given credentials

Create cursor

Execute query to fetch temperature, humidity, and timestamp records from "temperature_humidity" table

Store fetched information in a variable

Data Conversion and Preprocessing

Convert statistics to DataFrame with columns 'temperature', 'humidity', and 'timestamp'

Convert temperature and humidity columns to float type

Convert timestamp column to datetime format

Outline Characteristic and Target Variables

Set 'temperature' column as function variable (X)

Set 'humidity' column as target variable (y)

Train-test split for Humidity Prediction Model

Split data into training and testing sets with 80/20 split

Model Training and Prediction for Humidity

Initialize Linear Regression model

Train model using training statistics

Predict humidity values on test set

Calculate model score (R^2) and print it

Model Training and Prediction for Temperature (based on humidity)

Initialize some other Linear Regression model for temperature prediction

Train model using humidity data as input (reshaped as required) and temperature as target

predict temperature values on check set

Calculate model rating (R^2) and print it

Visualization with Multiple Subplots

Initialize figure with subplots

Subplot 1: Histogram of Temperature

Plot histogram for temperature data

add titles and labels

Subplot 2: Histogram of Humidity

Plot histogram for humidity data

add titles and labels

Subplot 3: Scatter plot of Temperature vs Humidity

Create scatter plot with temperature on X-axis and humidity on Y-axis

Set axis limits, titles, and labels

Subplot 4: Scatter plot of Humidity vs Temperature

Create scatter plot with humidity on X-axis and temperature on Y-axis

Set axis limits, titles, and labels

Subplot 5: Line plot of Actual vs Predicted Humidity
Plot actual humidity values and anticipated values at the equal plot
add title, labels, and legend

Subplot 6: Line plot of Actual vs Predicted Temperature
Plot actual temperature values and anticipated values at the same plot
add title, labels, and legend

Universal layout adjustment for subplots
Modify format to prevent overlap

Line Chart for Temperature and Humidity through the years
Initialize every other figure for time collection plot
Plot temperature values over timestamp
Plot humidity values over timestamp
add title, axis labels, legend, and grid

display all plots
display the figures with plots

Webpage

HTML file Structure:

- set up the document kind and language attributes.

Head section:

- Set character encoding to UTF-8 and viewport settings for cell compatibility.
- outline a name for the page as "HyGrow".
- upload inline CSS to style the page factors including font, colors, background, navbar, buttons, and footer.

frame Section:

- add a full-web page background with a blurred farm picture.

- Navbar:
- add a navbar with a link to the "Home" segment.

- Main Heading:
- display the title "HyGrow" with custom styling.

- Marquee:
- show a scrolling textual content welcoming users to HyGrow.

- Motor control Button:
- Create a button categorized "Flip Motor On" to toggle motor status.
- when clicked, it will call the `controlMotor()` function to send a request to control the motor.

- Status Display:
- show a placeholder text displaying "Motor Status: Unknown" to indicate motor or sensor status.

- Buttons for Temperature and Humidity Check:
- add buttons to check temperature and humidity values.
- each button, when clicked, will call respective functions `checkTemperature()` and `checkHumidity()` to fetch statistics.

- JavaScript for Interactive Functions:
- define `server_ip` variable for the server address.

- `controlMotor()` function:
- test contemporary button text to determine motor action ("on" or "off").
- send a fetch request to the server to toggle the motor state.

- update the button textual content and status display based on response.

- `checkTemperature()` function:
- send a fetch request to get temperature statistics.
- Update the status display with the temperature value.

- `checkHumidity()` function:
- send a fetch request to obtain humidity data.
- update the status display with the humidity value.

- `fetchData(action)` function:
- General function to fetch data based on specified action (temperature, humidity, etc.).
- update the status display with fetched data.

- Contact Section:
- display contact information with links to email and contact.

- Footer:
- show footer text with copyright information.

Mobile Application

Component: HomeScreen
- Import necessary modules:

- define HomeScreen Component:
- Return the primary layout inside a fragment

- Main Container (View):
- Apply `styles.container` for center alignment and background color.

- Logo Image:
- Use `Image` component to display the HyGrow logo.
- Set `source` attribute to local image path for the logo.

- Title Text:
- Display a title message "Welcome to HyGrow!".
- Apply `styles.title` for font length, colour, and alignment.

- Description Text:
- show a description of the HyGrow service, with line breaks.
- Apply `styles.description` for font length, colour, and alignment.

Component: Motor
- Import important modules:
- Define Motor Class Component (Extends React.Component):
- Properties:
- `url`: A string representing the base URL for motor manipulate movements.
- State (MotorState):
- `motorStatus`: Boolean to track whether the motor is ON or OFF.

- Constructor:
- Initialize `state` with `motorStatus` set to `false` (motor is OFF by default).
- Call `getMotorStatus` to fetch the current motor status from the server.

- Methods:
- `getMotorStatus`: Fetches current motor status from the server.
- Sends a request to the `url`.
- If response is "Motor On":
- Set `motorStatus` to `true` (motor is ON).
- If error occurs:
- Log error to the console and set `motorStatus` to `false`.

- `setMotorStatus`: Updates the motor status on the server.
- Accepts `sta` (string, either 'on' or 'off') to indicate desired motor state.
- Sends request to `url` with `sta` appended.
- Updates `motorStatus` in `state` based on response.
- Handles any errors through logging to the console.

- `toggleMotor`: Toggles motor status when `switch` is pressed.
- Accepts `value` (boolean) indicating whether motor need to be ON or OFF.
- Sets `motorStatus` in state to `value`.
- presents alert to user with the new motor status.
- Calls `setMotorStatus` with either 'on' or 'off' based on `motorStatus`.

- Render Method:
- Returns main layout inside a container `View` with `styles.container`.
- layout includes:
- Title `Text` element : Displays "Motor".
- Description `Text` element : short description.
- Motor Status `Text` element:
- Shows modern motor status ("Motor Status: ON/OFF").
- `Switch` element for toggling motor:
- Binds `onValueChange` to `toggleMotor`.
- Uses `motorStatus` from state as the value to control switch position.

- Define Interfaces:
- `MotorState`: Holds `motorStatus` boolean.
- `MyProps`: Placeholder interface for props (currently empty).

Component: Humidity

- Import vital modules:

- Define Humidity Class Component (Extends React.Component):
- Properties:
- `url`: A string that stores the server URL for fetching humidity data.
- `getHumidity` function:
- Fetch data from the humidity endpoint (`url`).
- If successful:
- Log "Successful" to the console.
- Set `humiditytxt` in the component's state to the fetched data (humidity).
- If error:
- Log "Failed to return humidity" and set `humiditytxt` to `'Undefined'`.
- Constructor:
- Initialize `state` with:
- `humiditytxt` as an empty string.
- Call `getHumidity()` to load initial humidity data.
- Render Method:
- Returns primary layout inside a fragment (`<>...</>`).

- Container (View):
- Title Text (using `styles.title`):
- display "Humidity" as the main title.
- Description Text :
- short description about checking humidity updates.

- Humidity Container :
- Humidity Display :
- Display current humidity text from `humiditytxt` state.
- Subtitle Text :
- Label "Current Humidity" below humidity percentage.
- Humidity Icon :
- Display icon image with URI link for humidity.
- Check Humidity Button (`TouchableOpacity`):

- display button classified "Check Humidity".
- On press, triggers `getHumidity` to refresh humidity records.

- Define Interfaces:
- `MyHumidityState`: Defines state with `humiditytxt` as a string.
- `MyProps`: Placeholder interface for props (currently empty).

CHAPTER 5

RESULTS AND DISCUSSION

A dedicated web page (Fig 5.1) and mobile application (Fig 5.2) have been developed to facilitate remote watering of plants and efficient management of watering schedules. Whether users are at home or away, this technology allows for precise and remote plant nurturing, thereby ensuring optimal growth.

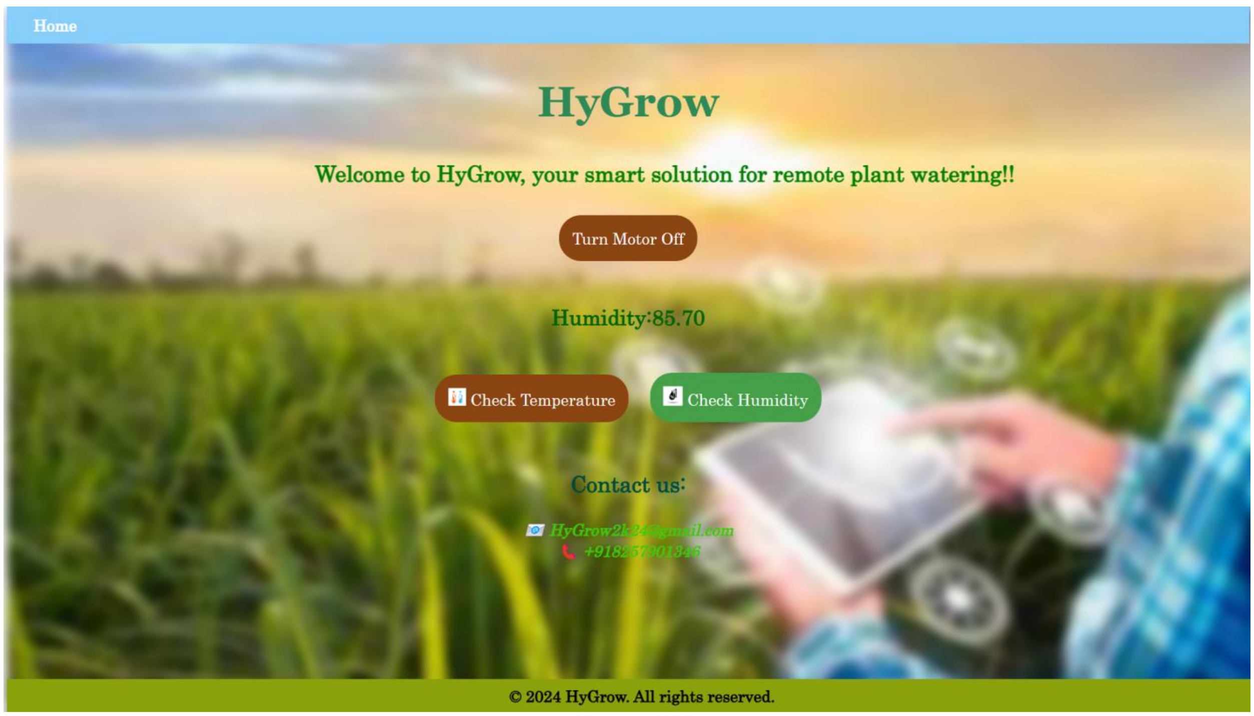

Fig 5.1 HyGrow Webpage

Fig 5.1 illustrates the webpage which serves the same purpose as mobile application.

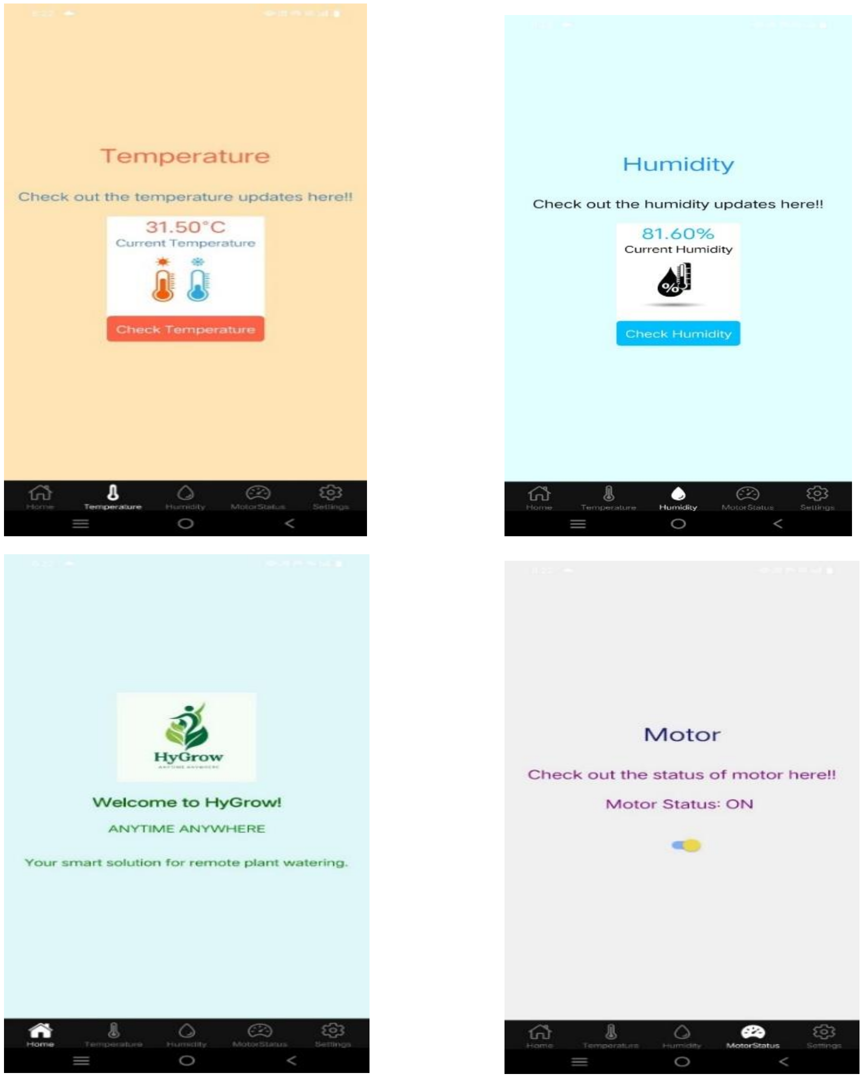

Fig 5.2 Screenshots of mobile app

Fig 5.2 illustrates screen shots of our mobile application in which users can access data remotely.

The system incorporates a machine learning algorithm using Linear Regression to predict humidity based on temperature data. This predictive capability enhances the system's ability to make informed decisions about watering schedules.

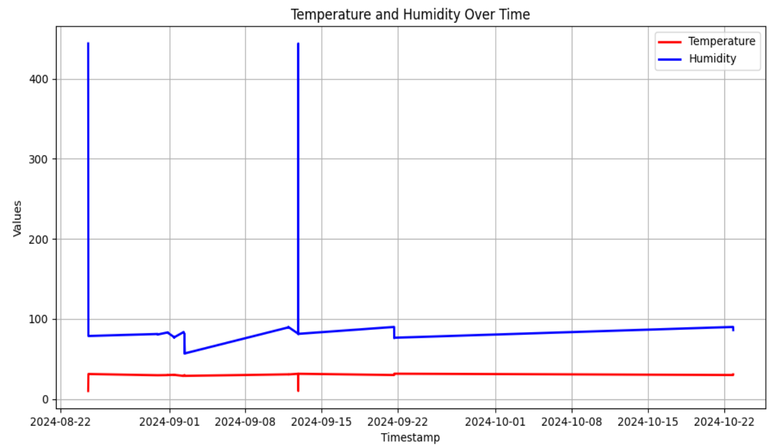

Fig 5.3 Measures temperature and humidity over time

Fig 5.3 measures temperature (red) and humidity (blue) over time. Humidity starts very high and then drops to around 100 units, stabilizing with an 80 to 100 unit swing. Temperature remains relatively stable with small fluctuations. The immediate drop in atmospheric humidity could be due to a change in weather conditions or re-calibration of equipment.

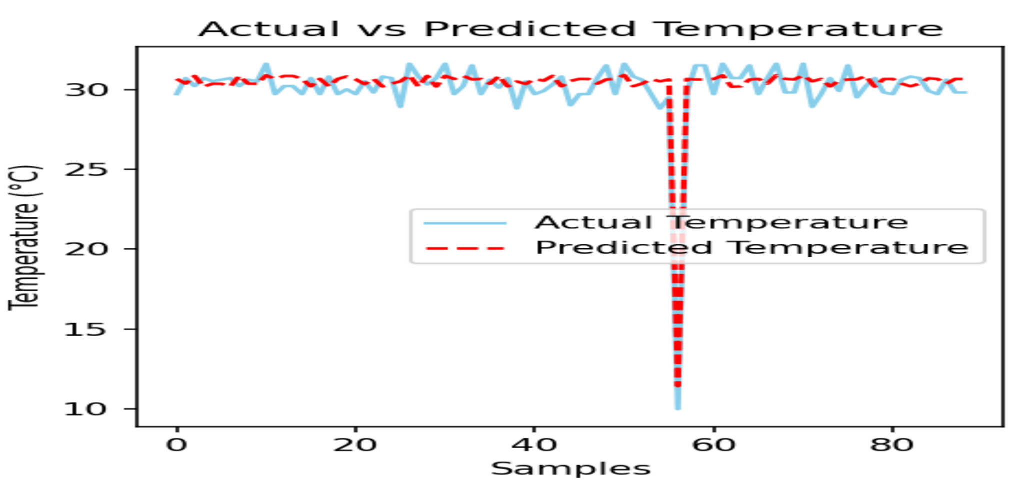

Fig 5.4 Actual vs Predicted Temperature

Fig 5.4 compares the actual temperature (solid light blue line) with the predicted temperature (dashed red line) for several samples. The actual temperature peaks at about 31°C, while the predicted temperature shows a smoother change, indicating that the model does not perform well during sudden changes.

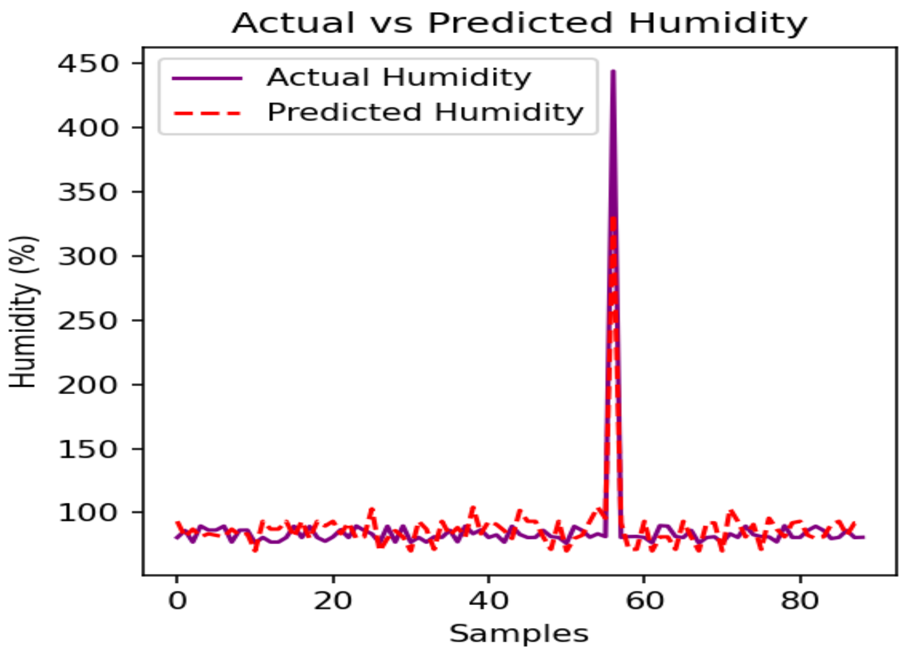

Fig 5.5 Actual vs Predicted Humidity

Fig 5.5 compares the actual soil moisture (solid red line) and the predicted soil moisture (dashed red line). The model tends to overestimate humidity levels, especially during rapid increases, indicating that it needs further tuning to handle extreme conditions.

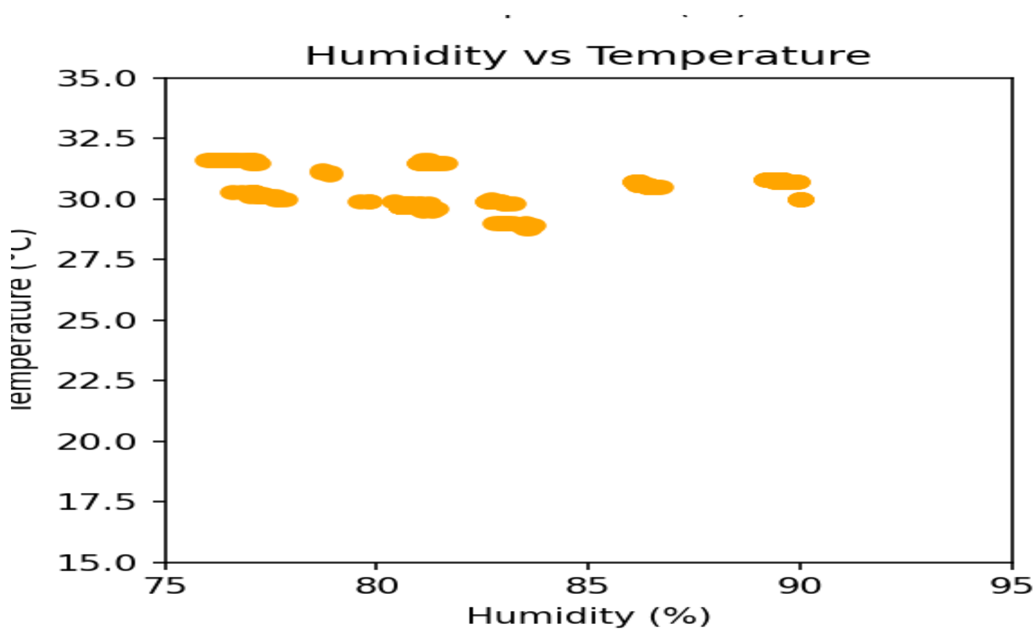

Fig 5.6 Humidity vs Temperature

Fig 5.6 depicts another scatter plot of humidity versus temperature. In contrast to the previous plot, this graph shows an inverse relationship between temperature and humidity. As temperature increases, humidity tends to decrease. This inverse proportion is consistent across the data range and provides valuable insights for climate modelling and understanding how temperature changes affect soil moisture.

CHAPTER 6

CONCLUSION AND FUTURE WORKS

6.1 CONCLUSION

The HyGrow project demonstrates the transformative potential of integrating IoT and machine learning technologies in agricultural practices, particularly in the irrigation management. By providing real-time monitoring, data analysis, and automated control, HyGrow offers a comprehensive solution to the persistent challenges of water inefficiency and soil degradation in agriculture. Key achievements of the project include:

- Development of a fully functional IoT based smart irrigation system
- Implementation of a machine learning model for predictive analysis of environmental conditions.
- Creation of user-friendly interfaces for remote monitoring and control, enhancing accessibility
- Demonstration of improved water efficiency and crop health in controlled tests
- Contribution to sustainable farming practices through optimized resource utilization.

The success of HyGrow highlights the significant potential of smart irrigation systems in revolutionizing agricultural practices. By leveraging real-time data and automated decision-making, our system not only ad dresses immediate concerns of water conservation but also paves the way for more sustainable and productive farming methods.

6.2 FUTURE WORKS

Looking forward, the HyGrow system presents numerous opportunities for further development and broader application:

- Integration with weather forecast data to further refine irrigation scheduling
- Expansion of the machine learning model to include crop-specific parameters and growth stage analysis
- Development of a scalable infrastructure to support larger agricultural operations
- Exploration of additional sensors to monitor other crucial environmental factors
- Collaboration with agricultural experts to fine-tune the system for various crop types and regional conditions

In conclusion, HyGrow represents a significant step towards addressing global food security challenges through innovative technology. By enhancing water efficiency, reducing labour requirements, and providing data driven insights, our system contributes to the broader goal of sustainable agriculture. As we continue to refine and expand this technology, we envision a future where smart irrigation systems play a crucial role in ensuring food security, conserving resources, and promoting environmental sustainability in agriculture worldwide.

CHAPTER 7

REFERENCES

Chavan, A. (2023). Smart Drip Irrigation System using IoT. Netafim India Blog.

Kanwal, H., Ahmad, I., Ahmad, A., & Li, Y. (2023). Yield Forecasting and Assessment of Interannual Wheat Yield Variability Using Machine Learning Approach in a Semi-Arid Environment. Journal of Agricultural Science & Technology.

El Mezouari, A., El Fazziki, A., & Sadgal, M. (2022). Smart Irrigation System. IFAC-PapersOnLine, 55, 75-80.

García, L., Parra, L., Jimenez, J., Lloret, J., & Lorenz, P. (2020). IoT-Based Smart Irrigation Systems: An Overview on the Recent Trends on Sensors and IoT Systems for Irrigation in Precision Agriculture. Sensors, 20, 1042.

Khaled, O., Yahya, A., Alfian, G., Tarolli, L.S., Lubis, M., & Pratama, A.A. (2022). An overview of smart irrigation systems using IoT. Journal of Agricultural Informatics, 13, 32-47.

Kumar, A., Ranjan, P., & Saini, V. (2022). Smart Irrigation System Using IoT. In: Agri-Food 4.0 (Advanced Series in Management, Vol. 27). Emerald Publishing Limited, pp.161-175.

Masson-Delmotte, V., et al. (2020). Climate Change and Land. Intergovernmental Panel on Climate Change, Geneva, Switzerland.Obaideen, K., Dhall, A., Sharma, S., Goel, S., & Bhatt, D. (2021). Recent advancements and challenges of Internet of Things in smart agriculture: A survey. Future Generation Computer Systems, 125, 822-838.

Raza Ahmad, M., Jamil, Y., Haq, Z., & Amin, N. (2021). An efficient portable and fast response digital humidity meter for use in the preservation of agricultural food products. Pakistan Journal of Agricultural Sciences.

Raza, M.A., Ashfaq, M., Hassan, S., & Hussain, I. (2021). Implication of irrigation reforms on wheat productivity: A case study of Punjab, Pakistan. Pakistan Journal of Agricultural Economics.

Sarwar, N., Maqsood, M., Mubeen, K., Shehzad, M., Bhullar, M.S., Qamar, R., & Akbar, N. (2010). Effect of different levels of irrigation on yield and yield components of wheat cultivars. Pakistan Journal of Agricultural Research, 47(3), 371-374.

Said Mohamed, E., Belal, A.M., Gad, G.A.A., & Abdelwahed, I. (2021). Smart Farming for Improving Agricultural Management. Egyptian Journal of Remote Sensing and Space Science, 24, 971-981.

Saraf, S.B., Gawali, D.H., Kulkarni, P.R., & Gundal, S. (2017). A Survey on Smart Agriculture Monitoring and Control System Using IoT. International Journal of Research and Scientific Innovation, 4(9), 83-86.

Vallejo-Gómez, D., Osorio, M., & Hincapié, C.A. (2023). Smart Irrigation Systems in Agriculture: A Systematic Review. Agronomy, 13, 342.

Sajjad, Y., Jaskani, M.J., Asif, M., & Qasim, M. (2021). Application of Plant Growth regulators in Ornamental Plants: A Review. Pakistan Journal of Agricultural Sciences.