Implementing IoT for Sustainable Agriculture and Livestock Health Monitoring

by

Shanjedul Tahmid
ID: CSE2002020006

Obaydulla
ID: CSE2103024069

Md.Iqbal Hossain
ID: CSE2103024022

Md.Naeem Hossain
ID: CSE2103024040

Md Amirul Alam Khan
ID: CSE1903018141

Md. Jalil Mia
ID: CSE2001019150

Supervised by
Md. Rashedul Islam

Submitted in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science and Engineering



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)

May 2025

APPROVAL

The thesis titled "Implementing IoT for Sustainable Agriculture and Livestock Health Monitoring" submitted by Shanjedul Tahmid (CSE2002020006), Obaydulla (CSE2103024069), Md.Iqbal Hossain (CSE2103024022), Md.Naeem Hossain (CSE2103024040), Md Amirul Alam Khan (CSE1903018141), Md. Jalil Mia CSE2001019150 to the Department of Computer Science and Engineering, Sonargaon University (SU), has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents.

Board of Examiners

Md. Rashedul Islam
Lecturer,
Department of Computer Science and Engineering
Sonargaon University (SU)

Supervisor

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 1

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 2

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 3

DECLARATION

We, hereby, declare that the work presented in this report is the outcome of the investigation performed by us under the supervision of **Md. Rashedul Islam, Lecturer** of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh. We reaffirm that no part of this thesis has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

(Md. Rashedul Islam)
Supervisor

Signature

Shanjedul Tahmid
ID: CSE2002020006

Obaydulla
ID: CSE2103024069

Md.Iqbal Hossain
ID: CSE2103024022

Md.Naeem Hossain
ID: CSE2103024040

Md Amirul Alam Khan
ID: CSE1903018141

Md. Jalil Mia
ID: CSE2001019150

ABSTRACT

This project presents the design and implementation of a smart agriculture and livestock health monitoring system based on Internet of Things (IoT) technology, developed to support farmers in rural and semi-urban areas. The system uses various sensors—including the DHT11 for temperature and humidity, MQ135 for detecting air quality, a soil moisture sensor, a rain detector, and the DS18B20 for monitoring animal body temperature—all connected through an ESP32 microcontroller. The data collected from the environment and animals is transmitted to cloud platforms like ThingSpeak and Google Sheets for real-time monitoring and data logging.

To assist in timely agricultural decisions, simple forecasting techniques are used in Python (via Google Colab) to predict next-day values of temperature, humidity, soil moisture, rainfall, and gas levels. These predictions are based on recent sensor data and help in making informed decisions regarding irrigation and farming activities. The system includes an automated irrigation control feature that activates the water pump based on both real-time and forecasted soil moisture readings, promoting efficient water use.

The system also checks whether current conditions are suitable for cultivating particular crops and identifies possible risks, such as high gas concentration or excessive humidity that could lead to fungal issues. For livestock health, it continuously tracks animal body temperature and sends instant alerts via Telegram to the medical team if any unusual readings are detected, allowing timely action and care.

This IoT-based solution offers a low-cost, automated, and practical system that can be easily used by small-scale farmers. It helps improve crop management, save resources like water, and ensure better livestock health, contributing to more sustainable and productive agriculture.

ACKNOWLEDGMENT

At the very beginning, we would like to express my deepest gratitude to the Almighty Allah for giving us the ability and the strength to finish the task successfully within the schedule time.

We are auspicious that we had the kind association as well as supervision of Md. Rashedul Islam, Lecturer, Department of Computer Science and Engineering, Sonargaon University whose hearted and valuable support with best concern and direction acted as necessary recourse to carry out our project.

We are also thankful to all our teachers during our whole education, for exposing us to the beauty of learning.

Finally, our deepest gratitude and love to my parents for their support, encouragement, and endless love.

LIST OF ABBREVIATIONS

API	Application Programming Interface
BMD,	Bangladesh Meteorological Department
BPH	Brown Planthopper
CNN	Convolutional Neural Network
ESP32	Espressif System-on-Chip 32
GDP	Gross Domestic Product
IoT	Internet Of Things
ML	Machine Learning
RH	Relative Humidity
SVM	Support Vector Machine
VWC	Volumetric Water Content
°C	Degree Celsius

TABLE OF CONTENTS

Title			Page No.
DECLARATION …….……..			iii
ABSTRACT ……...…….			iv
ACKNOWLEDGEMENT ……..……			v
LIST OF ABBREVIATION …….…….			vi
CHAPTER 1			1 – 5
INTRODUCTION			
1.1	Background and Motivation……..		1
1.2	Purpose and Goal of the Research…….		1-2
1.3	Why IoT and Machine Learning are Applied in Agriculture……		2-3
1.4	Organization of the Report……		3-4
1.5	Short Introduction About the System……		4-5
CHAPTER 2			6 – 11
LITERATURE REVIEW			
2.1	IoT-Based Agricultural Monitoring……...		6
2.2	Bangladesh-Specific Challenges in Agriculture……		6-7
	2.2.1	Climate Vulnerability and Unpredictable Weather……..	7
	2.2.2	Water Resource Management……..	7
	2.2.3	Soil Degradation and Fertility Issues……...	7
	2.2.4	Pest and Disease Outbreaks…….	7
	2.2.5	Financial Limitations and Technological Gaps……	8
	2.2.6	Infrastructure and Energy Deficiencies……...	8
	2.2.7	Need for Integrated Solutions……..	8
2.3	Machine Learning Applications in Farming…….		8
	2.3.1	Crop Yield Prediction……..……	9
	2.3.2	Disease Detection and Pest Control…….	9
	2.3.3	Smart Irrigation Management……..	9-10
	2.3.4	Benefits of Machine Learning in Agriculture……..	10

2.3.5 Challenges and Future Prospects……. 10-11

CHAPTER 3 12-24

SYSTEM DESIGN AND METHODOLOGY

3.1 Hardware Components……..……..…….. 12

3.1.1 ESP32 Microcontroller……..…….. 12-13

3.1.2 DHT11 Temperature and Humidity Sensor……... 13-14

3,1,3 MQ135 Gas Sensor……..…… 14-15

3.1.4 Rain Sensor……..……..…….. 15-16

3.1.5 DS18B20 Temperature Sensor……..……... 16-17

3.1.6 Soil Moisture Sensor……..…….. 17-18

3.1.7 Relay Module with Water Pump……..…… 18-19

3.2 Data Collection and Cloud Monitoring……..……... 19

3.2.1 ThingSpeak Platform Integration……. 19-10

3.2.2 Google Sheets for Historical Data Logging……. 20-21

3.3 Machine Learning Forecasting Model…….. 21

3.3.1 Next-Day Forecasting Techniques……... 21-22

3.3.2 Forecasted Parameters…… 22

3.4 Smart Irrigation Planning…….. 22-23

3.5 Risk Prediction and Disease Management…… 23

3.5.1 Fungal Infection Risk…… 23

3.5.2 Toxic Gas Risk and Farmer Advisory……. 23

3.5.3 Paddy Crop Suitability…….. 23

3.6 Alert System for Livestock Health…… 24

CHAPTER 4 25-42

IMPLEMENTATION AND RESULTS

4.1 Hardware Assembly…….. 25-26

4.2 Real-Time Monitoring and Data Visualization……. 26-27

4.3 Machine Learning Forecast Accuracy Evaluation……. 27-29

4.4 Automated Irrigation Control Outcomes……... 29-31

4.5 Livestock Monitoring Alerts……. 31-33

CHAPTER 5 34-42

CONCLUSION AND FUTURE WORK

5.1 Summary of Achievements……... 34-35

5.2 Limitations of the Current System…… 35-37

5.3 Future Improvements…… 37-39

5.3.1 Integration with External Meteorological Data……... 39-40

5.3.2 Advanced Data Visualization Tools……. 40-41

5.3.3 Development of Farmer-Friendly Mobile Applications…….. 41

5.3.4 Feature Expansion and System Scalability…….. 41-42

5.3.5 Interactive Dashboards and App Enhancements…….. 42

REFERENCES ……..…….. 43-44

LIST OF TABLES

Table No.	Title	Page No.
Table 1.1	Comparison between Traditional and Smart Agriculture Practices	2-3
Table 2.1	Key Agricultural Challenges in Bangladesh	8
Table 2.2	Key Machine Learning Applications in Agriculture	9-10
Table 3.1	ESP32 Specifications	12
Table 3.2	Basic Pin Functions of ESP32	13
Table 3.3	DHT11 Specifications	13.14
Table 3.4	Basic Pin Functions of DHT11	14
Table 3.5	MQ135 Specifications	14
Table.3.6	Basic Pin Functions of MQ135	15
Table.3.7	Rain Sensor Specifications	15
Table.3.8	Basic Pin Functions of Rain Sensor	16
Table.3.9	DS18B20 Specifications	16-17
Table.3.10	Basic Pin Functions of DS18B20	17
Table.3.11	Soil Moisture Sensor Specifications	17-18
Table.3.12	Basic Pin Functions of Soil Moisture Sensor	18
Table.3.13	Relay Specifications	18
Table.3.14	Basic Pin Functions of Relay Module	19
Table 3.15	Structure of Logged Data in Google Sheets	20-21
Table 3.16	Parameters Condition	22
Table 3.17	Placement: Smart Irrigation Decision	22
Table 3.18	Fungal risks based on humidity and temperatures	23
Table 3.19	Gas level readings from the MQ135 sensor are used to alert farmers about air toxicity	23
Table 4.1	Connection Overview	26
Table 4.2	A sample of predicted vs actual temperature	28-29

LIST OF FIGURES

Figure No.	Title	Page No.
Fig.1.1	ESP32 Microcontroller	12
Fig.1.2	DHT11 Sensor	13
Fig.1.3	MQ135 Sensor	14
Fig.1.4	Rain Sensor	15
Fig.1.5	DS18B20 Sensor	16
Fig.1.6	Soil Moisture Sensor	17
Fig.1.7	Relay Module and Water Pump	18
Fig.1.8	ThingSpeak dashboard showing live field data graphs	20
Fig.1.9	ML Forecasting Model Workflow	21
Fig 2	Hardware Diagram for Smart Agriculture System	25
Fig 2.1	Apply Forecasting with Linear Regression	28
Fig 2.2	Smart Irrigation Decision Flow	31
Fig 2.3	Body Temp Monitoring Flow with Telegram Interface	33

CHAPTER 1

INTRODUCTION

1.1 Background and Motivation

Agriculture is the backbone of many economies, especially in developing countries where a significant portion of the population depends on farming for their livelihood. However, traditional agricultural practices often suffer from inefficiencies, unpredictable weather conditions, water scarcity, and sudden disease outbreaks, leading to crop failures and financial hardships for farmers. The integration of modern technologies like the Internet of Things (IoT) [1] has opened new possibilities for revolutionizing agriculture by enabling real-time monitoring, predictive analysis, and automated decision-making.

IoT-based systems, when combined with smart sensors and cloud computing, allow continuous monitoring of critical environmental factors such as temperature, humidity, rainfall, soil moisture, and toxic gas levels. Devices like the ESP32 microcontroller [2] provide low-cost and efficient solutions for collecting and transmitting real-time data. Such technology empowers farmers to optimize irrigation, detect early signs of crop diseases, and manage livestock health effectively.

Motivated by the need to support financially struggling farmers, this project aims to develop an affordable, reliable, and intelligent farming assistance system. By predicting next-day weather conditions, irrigation needs, and potential cultivation risks, the system enables farmers to make proactive decisions, minimize resource wastage, and enhance crop yield. Furthermore, automated alerts for livestock health issues provide an additional layer of protection, ensuring comprehensive farm management.

In addition, incorporating local meteorological data, such as from the Bangladesh Meteorological Department (BMD) [3], and analyzing Volumetric Water Content (VWC) can further refine the accuracy of predictions. Ultimately, the goal is to make smart farming accessible and practical for farmers who need it most, improving their productivity, profitability, and resilience against uncertainties.

1.2 Purpose and Goal of the Research

The primary purpose of this research is to design and implement an Internet of Things (IoT)-based system that enhances agricultural practices by providing real-time environmental monitoring, intelligent forecasting, and automated farm management. Agriculture, particularly in developing regions, faces numerous challenges such as water scarcity, unpredictable rainfall, and disease outbreaks, which often lead to crop failure and financial losses for farmers [1]. By leveraging modern technologies such as IoT and cloud computing, the system aims to address these challenges through continuous data collection, analysis, and predictive decision-making.

Using a cost-effective ESP32 microcontroller [2] and a network of smart sensors, the proposed system gathers critical parameters including temperature, humidity, soil moisture,

rainfall, and toxic gas levels. This data is transmitted to cloud platforms like ThingSpeak and Google Sheets for real-time monitoring and historical analysis. Machine Learning (ML) models, developed using Python in Google Colab, utilize the collected data to forecast next-day weather conditions, soil moisture status, and potential cultivation risks. These forecasts assist farmers in planning irrigation schedules, applying fertilizers, and taking preventive actions against crop diseases and livestock health issues.

Furthermore, the system integrates local meteorological data, such as information from the Bangladesh Meteorological Department (BMD) [3], to improve the accuracy of environmental predictions. By focusing on affordability, ease of use, and actionable insights, the goal of this research is to empower financially struggling farmers to optimize resource utilization, enhance crop yield, protect livestock health, and ultimately achieve greater economic resilience in the face of agricultural uncertainties.

1.3 Why IoT and Machine Learning are Applied in Agriculture

Agriculture plays a vital role in feeding the world's growing population, yet it continues to face significant challenges, including climate variability, pest outbreaks, soil degradation, and inefficient resource management. Traditional agricultural practices often rely heavily on manual observation and historical patterns, which can lead to delayed responses and suboptimal yields [1]. To overcome these limitations, modern technologies such as the Internet of Things (IoT) and Machine Learning (ML) have emerged as transformative tools in agricultural practice.

IoT enables real-time monitoring of critical parameters such as soil moisture, temperature, humidity, toxic gas concentration, and rainfall through low-power devices like the ESP32 microcontroller [2]. These sensors collect continuous data from the field, which is then transmitted wirelessly to cloud platforms for storage and analysis. By providing farmers with live updates and trends, IoT empowers timely decision-making, optimizes water usage, and reduces crop loss due to environmental stress [5].

Meanwhile, ML algorithms offer predictive analytics capabilities that traditional methods cannot match. By training models using historical sensor data, ML can forecast next-day weather conditions, disease risks, irrigation needs, and potential yield outcomes. This predictive power enables farmers to proactively manage resources, schedule irrigation efficiently, and take preventive measures against pests and diseases [5].

Parameter	Traditional Agriculture	IoT based Agriculture
Decision Making	Based on experience and manual observation	Based on real-time data and predictive analytics
Resource Usage	High wastage of water and fertilizers	Optimized resource utilization
Risk Management	Reactive to crop diseases	Proactive prevention through

	and weather	prediction
Cost Efficiency	Higher operational costs	Reduced water, energy, and input costs

Table 1.1 Comparison between Traditional and Smart Agriculture Practices

Incorporating local meteorological data sources, such as from the Bangladesh Meteorological Department (BMD), and field-level Volumetric Water Content (VWC) readings, further enhances the reliability of predictions [6]. This hybrid approach ensures farmers have access to both macro- and micro-level information for better farm management.

By integrating IoT and ML, agriculture transitions from a reactive system to a proactive, data-driven ecosystem. These technologies not only optimize productivity but also empower financially struggling farmers to achieve sustainability, resilience, and improved profitability.

1.4 Organization of the Report

This report is structured into multiple well-defined chapters to ensure clarity, logical flow, and comprehensive coverage of the research work undertaken. Each chapter builds upon the previous sections, leading the reader progressively from the problem statement toward the final solution and conclusions.

The first chapter, Introduction, establishes the background and motivation for the research. It highlights the critical challenges faced by the agricultural sector, particularly in developing countries like Bangladesh, where financial constraints, unpredictable weather conditions, and disease outbreaks threaten farming sustainability. The introduction further elaborates on the purpose and goal of developing an IoT-based monitoring and prediction system that leverages machine learning to assist farmers. It emphasizes the significance of real-time monitoring, early risk prediction, and intelligent irrigation management as tools to enhance agricultural productivity and resilience.

The second chapter, Literature Review, provides an in-depth exploration of prior studies and technological advancements in related fields. It discusses the role of IoT in agriculture, the relevance of smart farming practices, and the adoption of cloud computing for real-time data monitoring. A focused review is presented on Bangladesh-specific agricultural challenges, including limited technological adoption, erratic weather patterns, and infrastructural gaps. Furthermore, the literature review examines recent developments in machine learning applications for farming, including crop disease prediction, irrigation scheduling, and yield forecasting. Comparative analysis between traditional and smart farming approaches is also included to underline the advantages of the proposed system.

The third chapter, System Design and Methodology, outlines the architecture of the IoT-based agricultural monitoring system. It details the hardware components such as the ESP32 microcontroller, various environmental sensors (temperature, humidity, soil moisture, gas detectors, rain sensors), and the setup for water pump control using relays. The chapter

further explains the cloud platforms employed, including ThingSpeak for real-time graphical data representation and Google Sheets for historical data storage and machine learning integration. The methodology adopted for next-day forecasting using Python-based models in Google Colab is elaborated, along with the decision-making logic for smart irrigation and risk alert systems.

The fifth chapter, Conclusion and Future Work, summarizes the overall achievements and findings of the research. It reflects on how the proposed system meets its objectives of improving agricultural management and supporting financially struggling farmers. This section also identifies limitations encountered during the study, such as sensor calibration challenges and constraints due to limited local datasets. Proposals for future improvements are discussed, including the integration of additional environmental sensors (e.g., soil pH, solar radiation sensors), the inclusion of external meteorological data sources such as the Bangladesh Meteorological Department (BMD), and the development of user-friendly mobile applications for better farmer engagement and decision making.

Finally, the References section lists all scholarly works, articles, and research papers cited throughout the report, formatted according to the academic guidelines followed in this document. An Abbreviations section is also included to clarify the meaning of all technical terms and acronyms used.

Through this structured organization, the report provides a logical and coherent narrative from the identification of the problem, exploration of existing solutions, development of the proposed system, evaluation of its effectiveness, and suggestions for future enhancements. This approach ensures that readers gain a thorough understanding of the project and its contributions to the field of smart agriculture.
system focuses on improving agricultural efficiency by enabling real-time environmental monitoring, predictive analytics, and automated decision support. Using the ESP32 microcontroller, the system collects continuous data from multiple sensors, measuring parameters such as temperature, humidity, soil moisture, rainfall detection, toxic gas concentration, and animal body temperature [1].

1.5 Short Introduction About the System

The proposed IoT-based smart farming Collected data is transmitted wirelessly to cloud platforms like ThingSpeak and Google Sheets for real-time visualization and historical storage. Python-based Machine Learning (ML) models analyze this data to forecast next-day weather conditions, predict irrigation needs, and assess cultivation and livestock risks. Automated alerts are generated to assist farmers in taking timely preventive actions, including medical team notifications for livestock health issues.

Additionally, the system cross-verifies current field conditions with ideal farming parameters, providing farmers with actionable insights on whether environmental conditions are suitable for paddy cultivation. Integration of external meteorological data from sources like the

Bangladesh Meteorological Department (BMD) and analysis of Volumetric Water Content (VWC) further strengthen the prediction reliability [2].

Overall, this affordable and user-friendly system is designed to empower financially struggling farmers with intelligent, data-driven tools for sustainable agricultural productivity.

CHAPTER 2

LITERATURE REVIEW

2.1 IoT-Based Agricultural Monitoring

The integration of the Internet of Things (IoT) has significantly enhanced agricultural practices by enabling real-time field monitoring and intelligent decision-making. IoT systems in agriculture typically consist of interconnected sensors and devices, supported by a microcontroller such as the ESP32, which is widely used for its low power consumption, Wi-Fi connectivity, and affordability [7]. These systems continuously collect data from the field, monitoring critical environmental parameters such as soil moisture, temperature, humidity, toxic gas concentrations, and rainfall.

Soil moisture sensors help determine the Volumetric Water Content (VWC) of the soil, allowing precise irrigation scheduling to avoid over-watering or drought stress. Environmental sensors like the DHT11 monitor temperature and humidity, while MQ135 sensors detect the presence of harmful gases such as ammonia and methane, indicating potential fertilizer misuse or livestock waste buildup. Rain sensors are employed to detect precipitation events, aiding in smart irrigation planning and reducing unnecessary water pump operation.

All collected data is transmitted wirelessly to cloud-based platforms, such as ThingSpeak, where farmers can monitor environmental trends through graphical interfaces. This real-time visibility enables farmers to detect early signs of water deficiency, disease outbreaks, or harmful gas accumulation and respond proactively. Simultaneously, data is also logged to platforms like Google Sheets, providing historical records that can be analyzed later using machine learning (ML) techniques for forecasting and deeper insights [8].

By continuously monitoring environmental and soil conditions, IoT-based agricultural systems allow farmers to optimize water use, minimize resource wastage, protect crop and livestock health, and ultimately increase yield stability. Moreover, these technologies are particularly beneficial for financially struggling farmers, as they provide affordable, scalable, and easy-to-deploy solutions that enhance farming productivity while reducing operational risks.

The adoption of IoT in agriculture thus marks a critical step toward transforming traditional, reactive farming methods into proactive, data-driven practices that are resilient against climate variability and resource constraints.

2.2 Bangladesh-Specific Challenges in Agriculture

Agriculture is a fundamental part of Bangladesh’s economy, contributing approximately 12.65% of the Gross Domestic Product (GDP) and employing over 40% of the total workforce [9]. Despite its importance, the sector faces numerous challenges that threaten

productivity, food security, and farmers' livelihoods. These challenges arise from environmental, socio-economic, technological, and infrastructural factors that are often more severe compared to global standards.

2.2.1 Climate Vulnerability and Unpredictable Weather

Bangladesh is one of the most climate-vulnerable countries globally due to its geographic location in the delta of three major rivers and its low-lying coastal plains [10]. Floods, cyclones, droughts, and erratic rainfall patterns severely impact agricultural productivity. For instance, excessive flooding can destroy standing crops, while sudden droughts reduce soil moisture content, affecting seed germination and plant growth.

The unpredictability of rainfall patterns, intensified by climate change, makes traditional agricultural scheduling based on historical data unreliable [11]. Farmers often face difficulties in correctly timing planting and harvesting activities, leading to lower yields and increased vulnerability to food insecurity.

2.2.2 Water Resource Management

Efficient water management is another pressing issue. In Bangladesh, around 80% of freshwater withdrawal is dedicated to irrigation, mainly for rice cultivation [12]. However, unsustainable groundwater extraction, inefficient irrigation methods, and pollution from agrochemical runoff are degrading the quality and availability of water resources.

Due to the absence of modern irrigation systems integrated with soil moisture sensors, farmers still rely on manual judgment, leading to water wastage and salinization of soil. Adoption of real-time IoT-based irrigation management could significantly enhance water use efficiency.

2.2.3 Soil Degradation and Fertility Issues

Continuous monoculture cropping patterns, particularly involving rice, and excessive use of chemical fertilizers have led to severe soil degradation across Bangladesh [13]. Salinity intrusion, especially in coastal zones, impacts more than 1 million hectares of arable land [14].

Monitoring critical soil health parameters such as Volumetric Water Content (VWC), pH levels, and organic matter content remains limited. Precision agriculture technologies have not yet been widely adopted, restricting the ability to reverse soil fertility decline.

2.2.4 Pest and Disease Outbreaks

Frequent pest outbreaks, including Brown Planthopper (BPH) infestations on rice and fungal diseases like rice blast, significantly reduce yields [15]. Farmers typically identify such threats visually after substantial damage has already occurred.

There is a lack of early detection systems using IoT sensors or Machine Learning (ML) based predictive models that could enable proactive pest and disease management. Consequently, farmers rely on excessive pesticide use, harming both crop health and the environment.

2.2.5 Financial Limitations and Technological Gaps

Most farmers in Bangladesh operate on small landholdings (less than two hectares) and lack the capital necessary to invest in advanced agricultural technologies [16]. Even where low-cost IoT solutions are available, lack of training and technical literacy hinder their effective utilization.

Furthermore, governmental incentives and subsidies for technology adoption remain inadequate, slowing the transition towards smart farming solutions.

2.2.6 Infrastructure and Energy Deficiencies

Reliable electricity supply and robust rural infrastructure are crucial for supporting agricultural technologies. However, rural Bangladesh often faces frequent power outages, particularly during the monsoon season [17].

Poor road connectivity and weak logistics infrastructure further hinder the timely supply of agricultural inputs and the marketing of produce, exacerbating post-harvest losses.

Challenge Area	Impact	Technology Solution
Climate Vulnerability	Unpredictable floods, droughts, crop loss	Real-time weather monitoring, ML-based forecasting
Water Management	Water wastage, groundwater depletion	IoT-enabled smart irrigation
Financial Constraints	Limited technology adoption	Low-cost IoT solutions, government subsidies

Table 2.1: Key Agricultural Challenges in Bangladesh

2.2.7 Need for Integrated Solutions

Considering these multifaceted challenges, a comprehensive agricultural system integrating IoT-based real-time environmental monitoring, ML-driven forecasting models, and automated decision-making frameworks is essential. Such integrated systems can provide actionable insights, reduce operational costs, and improve resilience among financially vulnerable farmers.Efforts must also focus on enhancing farmers' digital literacy and creating affordable, scalable models tailored specifically to Bangladesh's socio-economic context.

2.3 Machine Learning Applications in Farming

The integration of Machine Learning (ML) in agriculture has marked a major shift towards smarter and more efficient farming practices. ML enables systems to learn from historical agricultural data, recognize patterns, and make informed predictions or decisions without the need for explicit programming [18]. This advancement is crucial in addressing challenges such as climate change, resource depletion, pest outbreaks, and the rising global demand for food.

2.3.1 Crop Yield Prediction

Accurate crop yield prediction is essential for ensuring food security and effective farm management. Traditionally, yield estimation was based on manual surveys and historical trends, often leading to inaccurate forecasts. ML models like Random Forest and Support Vector Machine (SVM) have enhanced yield prediction by utilizing environmental variables such as soil properties, rainfall, temperature, and crop history [1]. Predictive models enable farmers to plan sowing, harvesting, and marketing strategies more efficiently, ultimately leading to reduced losses and improved productivity.

2.3.2 Disease Detection and Pest Control

Plant diseases and pests are responsible for considerable agricultural losses worldwide. ML technologies, especially Convolutional Neural Networks (CNN), have been successfully employed in identifying plant diseases through image recognition [19]. By capturing images of plant leaves and analyzing them through trained ML models, farmers can detect early symptoms of diseases like bacterial blight, powdery mildew, or fungal infections. Early detection allows timely intervention, minimizing crop damage and reducing excessive pesticide use.

Moreover, pest infestation patterns can also be predicted using ML by analyzing factors such as humidity, temperature, and previous pest outbreak records. These predictive insights help in proactive pest management, reducing the reliance on broad-spectrum chemical sprays.

2.3.3 Smart Irrigation Management

Efficient water management is critical in agriculture, especially in water-scarce regions. ML algorithms optimize irrigation schedules based on real-time soil moisture data, weather forecasts, and crop water requirements. Systems equipped with ML can predict the precise amount of water needed at different stages of crop growth, thus conserving water and preventing both over-irrigation and under-irrigation [20].

For example, Decision Tree algorithms analyze historical sensor data and weather patterns to automate irrigation systems, ensuring that water is only used when necessary. This intelligent irrigation approach significantly improves water-use efficiency, especially important for countries like Bangladesh where water scarcity can impact agricultural output.

Application Area	Machine Learning Use
Crop Yield Prediction	Predicting expected harvests based on weather and soil data
Disease Detection	Early identification of plant diseases through image processing
Pest Management	Predicting pest outbreaks using climate and environmental data

Smart Irrigation	Optimizing watering schedules with soil moisture and weather inputs

Table.2.2 Key Machine Learning Applications in Agriculture

2.3.4 Benefits of Machine Learning in Agriculture

The integration of Machine Learning (ML) into farming brings numerous significant benefits. Firstly, it leads to increased efficiency by enabling automation and providing accurate predictions, thereby minimizing resource wastage. ML models can analyze large volumes of data quickly, allowing farmers to respond rapidly to changing field conditions, optimize the use of inputs like fertilizers and water, and ultimately enhance operational efficiency.

Moreover, higher productivity is achieved as early interventions based on ML-driven insights allow for timely pest control, disease management, and resource allocation. This results in better crop health and improved yields compared to traditional farming methods. In addition, ML contributes to cost reduction through intelligent resource management. By accurately predicting irrigation needs and identifying areas requiring intervention, farmers can reduce unnecessary expenditure on water, fertilizers, and pesticides, leading to more financially sustainable practices.

Environmental sustainability is another critical benefit of ML in agriculture. The use of data-driven decisions minimizes excessive chemical applications and promotes smarter water usage, thereby reducing the environmental footprint of farming activities. These benefits collectively demonstrate how ML not only improves farming productivity but also supports sustainable agricultural development.

2.3.5 Challenges and Future Prospects

Despite the promising advantages, the application of ML in agriculture faces several challenges. One major obstacle is limited data availability, particularly in countries like Bangladesh. The scarcity of high-quality, localized agricultural datasets can significantly hinder the accuracy and performance of ML models. Reliable data is essential for training and validating predictive models, and without it, the system's effectiveness may be compromised.

Another challenge is the high implementation cost associated with setting up sensor networks, data acquisition systems, and ML infrastructures. These expenses can be prohibitive for small-scale farmers, who often operate under tight financial constraints. In addition, there exists a technical knowledge gap among the farming community. Most farmers require extensive training and support to effectively utilize ML-based tools and interpret the insights generated by such systems.

However, the future prospects of ML in agriculture remain optimistic. With increasing government support, the development of affordable Internet of Things (IoT) devices, and initiatives aimed at building localized datasets, these challenges are gradually being

addressed. As technologies become more accessible and farmer education improves, ML has the potential to revolutionize agricultural practices by enhancing productivity, promoting sustainability, and improving the overall resilience of farming systems.

CHAPTER 3

SYSTEM DESIGN AND METHODOLOGY

3.1 Hardware Components

This section outlines the key hardware components used in the IoT-based agricultural monitoring system. Each device was selected to ensure low cost, energy efficiency, and high reliability for real-time environmental and livestock monitoring.

3.1.1 ESP32 Microcontroller

The ESP32 microcontroller serves as the core processing unit of the system. It handles sensor data acquisition, local processing, and wireless data transmission to cloud platforms.

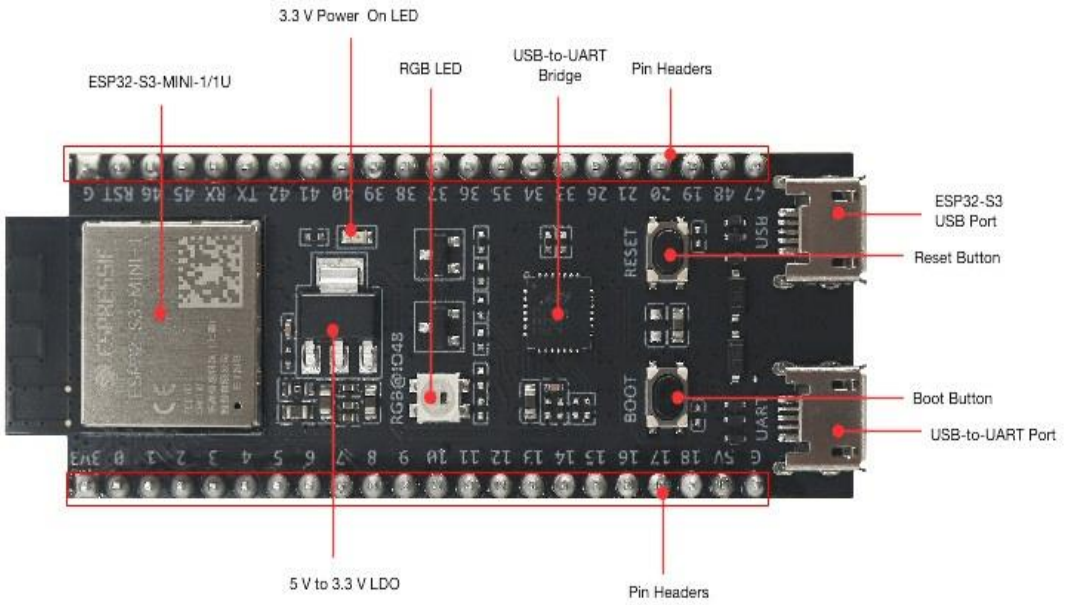


Figure 1.1 ESP32 Microcontroller

Feature	Description
Processor	Dual-core Tensilica LX6
Clock Speed	Up to 240 MHz
SRAM	520 KB
Flash Memory	4 MB
Connectivity	Wi-Fi 802.11 b/g/n, Bluetooth v4.2 BR
Operating Voltage	2.2 V to 3.6 V
GPIO Pins	36 total (25 usable)

Table 3.1: ESP32 Specifications

Pin	Function
GPIO0	Boot (used for flashing)
GPIO1	TX (Serial communication)
GPIO3	RX (Serial communication)
GPIO4	GPIO or external peripherals
GPIO5	GPIO or I2C (SCL or SDA)
GPIO12	SPI (MISO) or GPIO
GPIO13	SPI (MOSI) or GPIO
GPIO14	SPI (SCK) or GPIO
GPIO15	Boot, also used for GPIO

Table 3.2 Basic Pin Functions of ESP32

3.1.2 DHT11 Temperature and Humidity Sensor

The DHT11 sensor is used to monitor ambient temperature and relative humidity in real time.

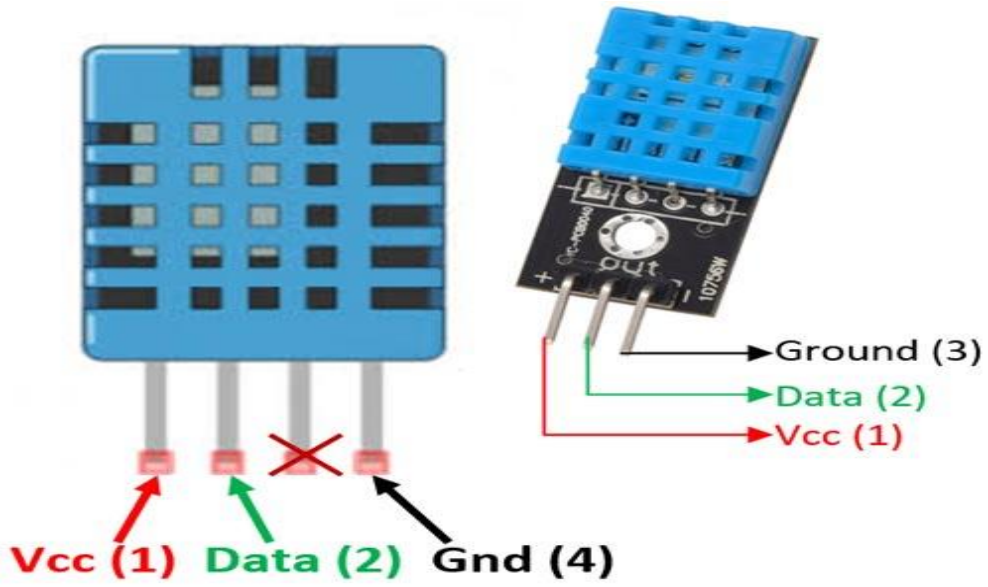


Figure 1.2 DHT11 Sensor

Feature	Description
Temperature Range	0°C to 50°C
Humidity Range	20% to 90% RH
Accuracy	±2°C and ±5% RH

Operating Voltage	3.3 V to 5.5 V
Interface	Single-wire digital signal

Table 3.3 DHT11 Specifications

Pin No	Function
Pin 1	VCC (Power)
Pin 2	Data
Pin 3	Not connected
Pin 4	GND (Ground)

Table 3.4: Basic Pin Functions of DHT11

3.1.3 MQ135 Gas Sensor

The MQ135 sensor detects hazardous gases like ammonia and methane, useful for identifying potential fertilizer misuse or livestock waste accumulation.

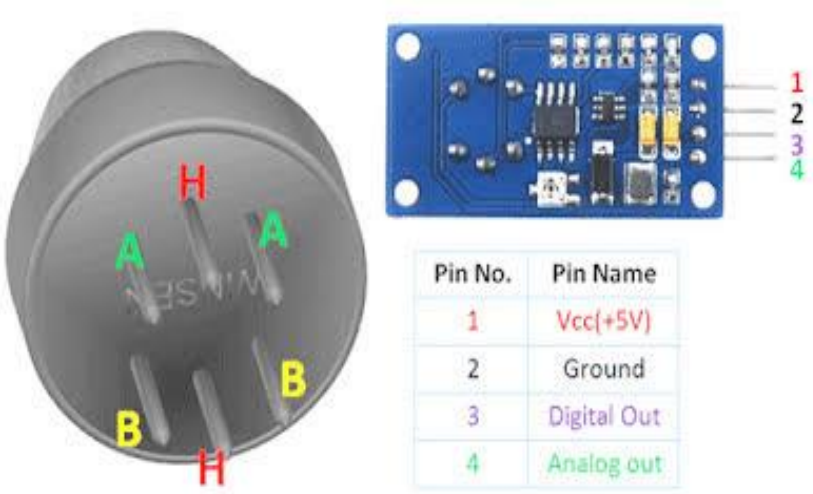


Figure 1.3 MQ135 Sensor

Feature	Description
Detectable Gases	Ammonia, Benzene, Alcohol, etc.
Operating Voltage	5 V
Output Type	Analog and digital
Warm-up Time	24 hours (initial use)

Table 3.5: MQ135 Specifications

Pin No	Function
Pin 1	AOUT (Analog)
Pin 2	DOUT (Digital)
Pin 3	GND
Pin 4	VCC

Table 3.6: Basic Pin Functions of MQ135

3.1.4 Rain Sensor

The rain sensor detects precipitation, helping the system decide whether irrigation is necessary.



Figure 1.4 Rain Sensor

Feature	Description
Output	Analog and digital
Operating Voltage	3.3 V to 5 V
Sensitivity	Adjustable via potentiometer

Table 3.7: Rain Sensor Specifications

Pin No	Function
Pin 1	AOUT (Analog)
Pin 2	DOUT (Digital)
Pin 3	GND
Pin 4	VCC

Table 3.8: Basic Pin Functions of Rain Sensor

3.1.5 DS18B20 Temperature Sensor

The DS18B20 sensor provides accurate body temperature monitoring for livestock, aiding in disease prediction.

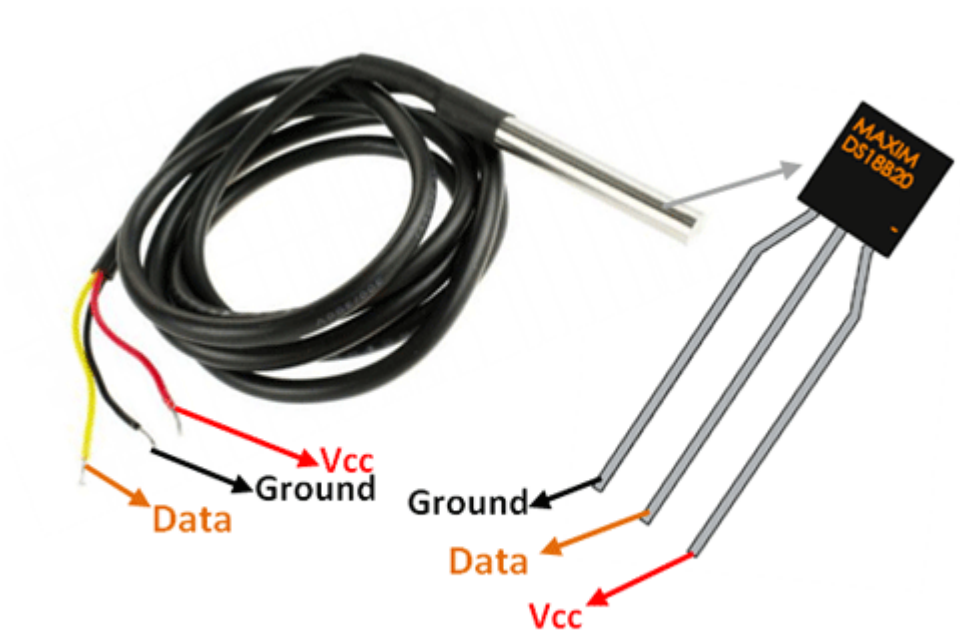


Figure 1.5 DS18B20 Sensor

Feature	Description
Temperature Range	-55°C to +125°C
Accuracy	±0.5°C
Communication	1-Wire Protocol

Feature	Description
Temperature Range	-55°C to +125°C
Operating Voltage	3.0 V to 5.5 V

Table 3.9: DS18B20 Specifications

Pin No	Function
Pin 1	GND
Pin 2	DATA
Pin 3	VDD(Power)

Table 3.10: Basic Pin Functions of DS18B20

3.1.6 Soil Moisture Sensor

This sensor determines soil water content to help manage irrigation intelligently.



Figure 1.6 Soil Moisture Sensor

Feature	Description
Output	Analog

Feature	Description
Output	Analog
Voltage Range	3.3 V to 5 V
Sensitivity	High for moist soil

Table 3.11: Soil Moisture Sensor Specifications

Pin No	Function
Pin 1	A0 (Analog Out)
Pin 2	VCC
Pin 3	GND

Table 3.12: Basic Pin Functions of Soil Moisture Sensor

3.1.7 Relay Module with Water Pump

The relay controls the pump to activate irrigation automatically when the soil moisture drops below a threshold.



Figure 1.7 Relay Module and Water Pump

Feature	Description
Operating Voltage	5 V
Switching Voltage	Up to 250 VAC or 30 VDC
Control Pin	Digital GPIO (ESP32)

Table 3.13 Relay Specifications

Pin No	Function
IN	Control from ESP32
VCC	Power Supply
GND	Ground
NO	Normally Open
COM	Common Contact

Table 3.14 Basic Pin Functions of Relay Module

3.2 Data Collection and Cloud Monitor

Efficient and reliable data collection is critical to the success of any IoT-based agricultural monitoring system. In this project, two cloud-based platforms have been utilized for data logging, visualization, and further machine learning (ML) analysis: ThingSpeak and Google Sheets. ThingSpeak provides real-time environmental data logging and monitoring capabilities, while Google Sheets is used to store historical data in a structured manner suitable for ML model training and prediction.

3.2.1 ThingSpeak Platform Integration

ThingSpeak is an open-source Internet of Things (IoT) platform that enables real-time data aggregation, storage, visualization, and analysis. The system is configured to send sensor data—including temperature, humidity, gas level, soil moisture level, and rainfall status—at one-minute intervals using a private channel secured with API keys. The ESP32 microcontroller handles the Wi-Fi connectivity and sends HTTP requests to ThingSpeak using the provided API keys.

The platform is organized with a single private channel that receives data from the following sensors:

- DHT11 for temperature and humidity.
- MQ135 for gas concentration.
- Soil moisture sensor for water level.
- Rain sensor for precipitation detection.

Each sensor reading is sent using separate fields on the ThingSpeak channel. The configuration allows real-time data plotting, threshold alerting, and external HTTP-based automation for other connected services.

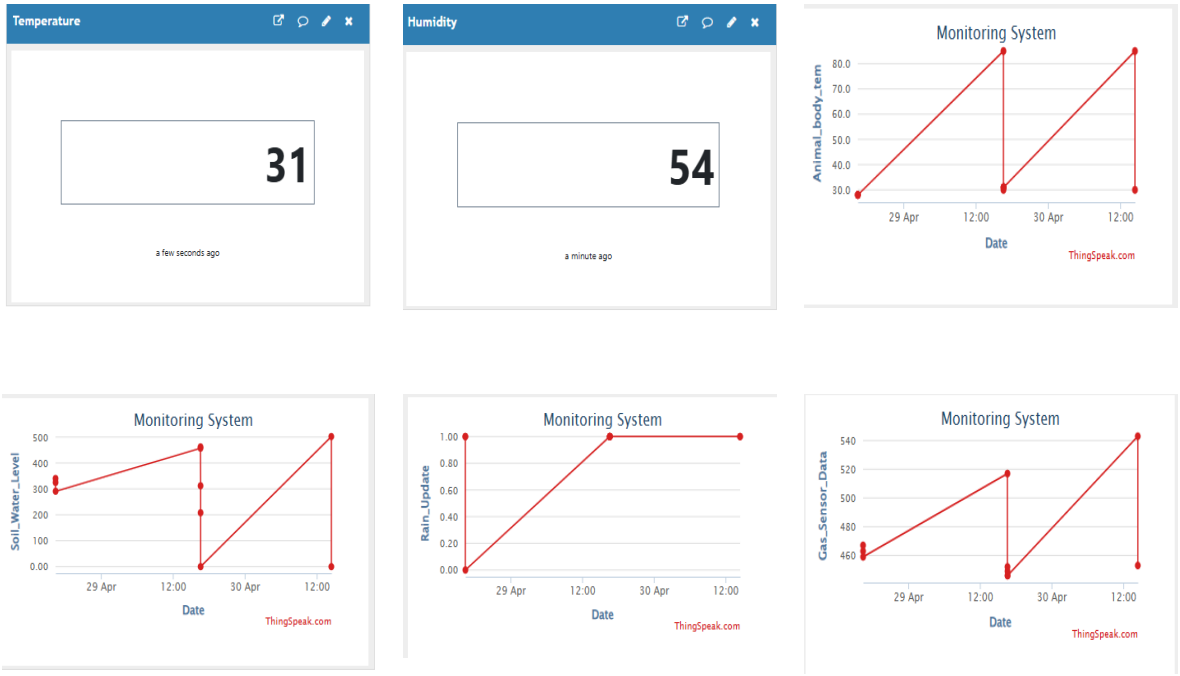


Figure 1.8 ThingSpeak dashboard showing live field data graphs

This integration allows researchers and farmers to monitor environmental parameters remotely and react promptly to unfavorable conditions [21].

3.2.2 Google Sheets for Historical Data Logging

To maintain a historical dataset for training machine learning models, data is simultaneously logged into Google Sheets using a Google Apps Script. The ESP32 transmits data through Wi-Fi to a Google Web App endpoint which then appends it to a spreadsheet hosted on Google Drive.

The logging frequency is approximately every five minutes, though this interval can be dynamically adjusted based on requirements. The spreadsheet includes both raw sensor readings and derived status indicators (e.g., “High”, “Low”, “Normal”) for enhanced classification. Table 6 shows the structure of the logged dataset.

Column Name	Description
Date	Date of data entry
Time	Time of data entry
Humidity	Percentage of atmospheric moisture
Humidity Status	Interpreted status (e.g., High/Normal/Low)
Temperature	Temperature in °C

Temperature Status	Status label based on thresholds
Gas	Gas concentration level
Gas Status	Categorization of gas risk level
Rain Status	Indicates rainfall detection (Yes/No)
Soil Water Level	Soil moisture sensor value
Soil Status	Soil condition (Wet/Dry)

Table 3.15 Structure of Logged Data in Google Sheets

This structured and labeled dataset provides an excellent foundation for ML modeling using platforms such as Google Colab. The collected data is used for next-day forecasting of temperature, humidity, rainfall, toxic gas presence, and soil moisture levels using supervised learning techniques [22].

Using both real-time (ThingSpeak) and long-term (Google Sheets) data platforms ensures robust system functionality, facilitates multi-source validation, and enhances the accuracy of predictions derived from ML models.

3.3 Machine Learning Forecasting Model

In this system, a machine learning (ML)-based approach was implemented to forecast next-day environmental parameters that directly affect agriculture, such as temperature, humidity, soil moisture, gas levels, and rainfall. The historical data is collected from Google Sheets and updated every few minutes using Apps Script. A lightweight forecasting model based on Linear Regression was selected due to its low computational complexity and suitability for resource-constrained systems.

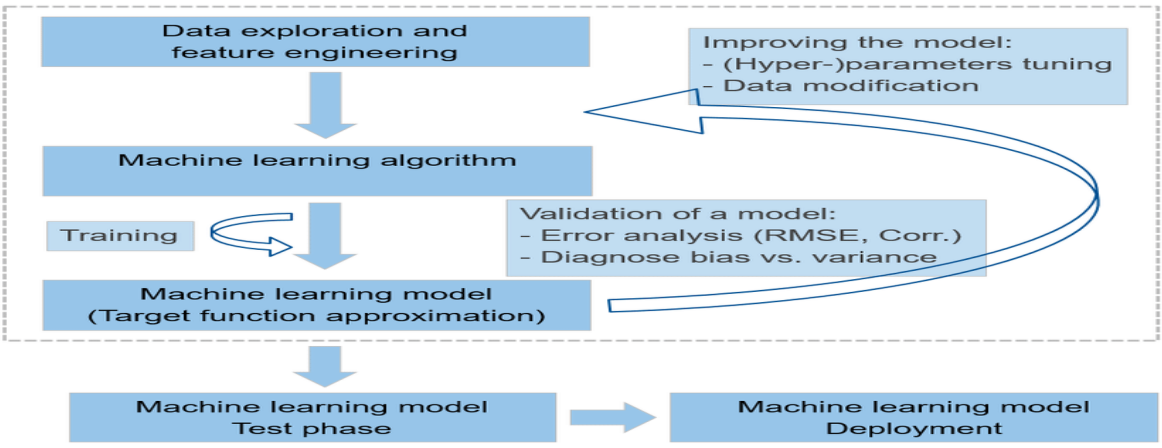


Figure 1.9 ML Forecasting Model Workflow

3.3.1 Next-Day Forecasting Techniques

The prediction model uses the last four days of historical data for each parameter. These values are numerically indexed and passed into a LinearRegression model from the scikit-

learn library. A separate model is trained for each environmental factor. Once trained, each model predicts the next sequential value for the coming day.

Mathematically, for any time-indexed sequence xtx_txt, the next predicted value xt+1x_{t+1}xt+1 is estimated as.

where θ0\theta_0θ0 and θ1\theta_1θ1 are learned parameters using least squares fitting.

3.3.2 Forecasted Parameters

The following parameters are predicted.

Parameter	Unit	Use Case
Temperature	°C	Crop suitability, animal health
Humidity	%	Fungal risk, disease detection
Rain Status	Boolean	Irrigation decision
Gas Level	ppm	Toxic risk to crops/livestock
Soil Moisture	%	Irrigation planning

Table 3.16 Parameters Condition

After prediction, a logic-based analysis converts these numerical predictions into qualitative statuses.This allows farmers to interpret the output easily and take proactive measures.

3.4 Smart Irrigation Planning

The irrigation system uses predicted soil moisture and rainfall data to determine the need for watering and to optimize pump usage. The logic is as follows.

- If soil is **Wet**, pump remains **OFF**.
- If soil is **Dry**, irrigation frequency and duration depend on the moisture level.

Soil Water Level	Times per Day	Duration per Cycle (min)
≤ 20	3	30
21–35	2	20
36–50	1	15
> 50	1	10

Table 3.17 Placement: Smart Irrigation Decision

This approach ensures minimal water wastage and power consumption, which is critical for financially constrained farmers. The system automatically calculates total minutes per day.

3.5 Risk Prediction and Disease Management

This section incorporates AI-based logic to identify environmental conditions that may lead to fungal infections, gas poisoning, or poor suitability for paddy cultivation. The system checks historical and forecasted values and highlights potential risks.

3.5.1 Fungal Infection Risk

Fungal risks are evaluated based on high humidity and moderate temperatures.

Condition	Risk Level
20°C–30°C & Humidity ≥ 85%	High
≥18°C & Humidity ≥ 70%	Moderate
Else	Low

Table 3.18 Fungal risks are evaluated based on high humidity and moderate temperatures

3.5.2 Toxic Gas Risk and Farmer Advisory

Gas level readings from the MQ135 sensor are used to alert farmers about air toxicity.

Gas (ppm)	Risk Level	Advice
>700	Very High	Ventilate immediately. Avoid animal exposure
501–700	High	Improve airflow and monitor fertilizer use
351–500	Moderate	Maintain ventilation and review practices.
≤350	Low	No action needed

Table 3.19 Gas level readings from the MQ135 sensor are used to alert farmers about air toxicity

These alerts are text-based and intended for low-literacy users.

3.5.3 Paddy Crop Suitability

A logic-based model determines if the environmental conditions are favorable for paddy cultivation.

- Temperature: 25–35°C
- Humidity: 70–90%
- Soil Moisture: 30–60
- Gas Level: <450 ppm

3.6 Alert System for Livestock Health

The livestock health alert is based on body temperature data from the DS18B20 sensor mounted on the animal. This sensor records body temperatures periodically 38.5–39.5C.

The Arduino system connected to the DS18B20 reads the temperature continuously. If abnormal values are found, it sends a formatted message via a Telegram bot to the registered vet or medical team. This is done using the Telegram Bot API and HTTP request logic from Arduino.

CHAPTER 4

IMPLEMENTATION AND RESULTS

4.1 Hardware Assembly

The hardware assembly forms the backbone of the proposed smart agriculture system. The core processing unit is the ESP32 microcontroller, which interfaces with a range of environmental and livestock monitoring sensors. Each sensor plays a critical role in data acquisition and control logic.

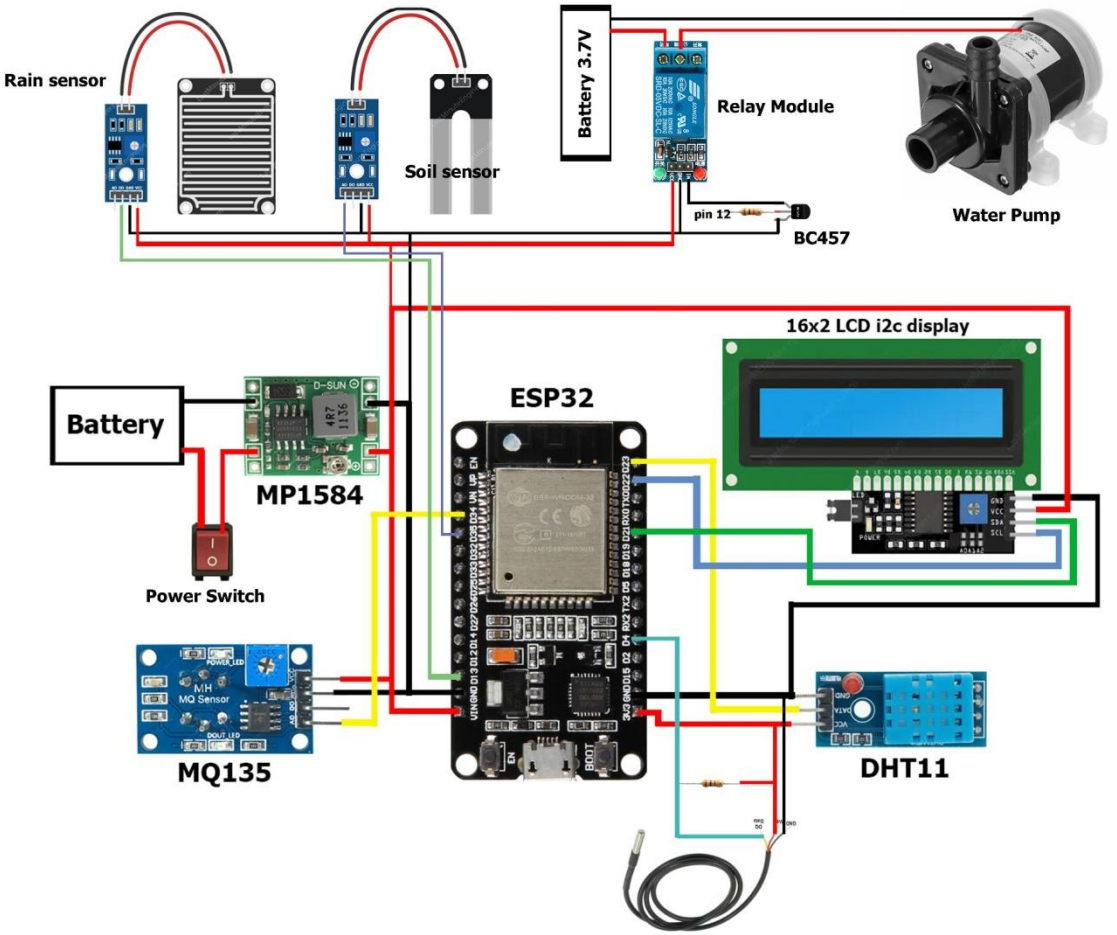


Figure 2. Hardware Diagram for Smart Agriculture System

This diagram clearly shows the pin mappings and power lines for all modules.

Prototype Structure: All components are mounted on a breadboard or PCB, enclosed in a weather-resistant casing when deployed in the field. Wire management and labeling ensure modular debugging and updates.

Component	Connection to ESP32	Interface Type
ESP32		Wi-Fi, GPIO, ADC, I2C
DHT11	GPIO(D4)	Digital (Single-wire)
MQ135	ADC(A0)	Analog
Rain Sensor	ADC(A3)	Analog
Soil Moisture Sensor	ADC(A2)	Analog
DS18B20	GPIO(D5)	Digital (OneWire)
Relay Module	GPIO(D12)	Digital (Trigger Input)
Water Pump	Relay Controlled	
LCD Display	SDA (D21), SCL (D22)	12C
MP1584 Regulator	Power Input	
Power Switch	Inline with Power Source	

Table 4.1 Connection Overview

4.2 Real-Time Monitoring and Data Visualization

The smart agriculture system integrates real-time data monitoring using both the ThingSpeak cloud platform and Google Sheets for transparent visualization and remote access. The ESP32 microcontroller serves as the central data acquisition unit, continuously gathering sensor readings including temperature, humidity, toxic gas levels, rainfall, soil moisture, and livestock body temperature. These values are updated at regular intervals: every one minute for ThingSpeak and approximately every five minutes for Google Sheets using Google Apps Script automation.

On ThingSpeak, each parameter is visualized on a private channel consisting of individual line charts that represent time-series data. The temperature and humidity are plotted together to assist in identifying microclimatic conditions, whereas toxic gas levels are independently visualized to track potential environmental hazards. The rainfall sensor readings are shown as binary or analog values depending on wet/dry conditions, while soil moisture levels are presented to guide irrigation decisions. Additionally, the DS18B20 temperature readings for livestock body health are recorded and visualized to detect anomalies.

For historical data tracking and machine learning forecasting, the same readings are also logged to a structured Google Sheet. This sheet includes timestamped columns for all measured parameters and their evaluated statuses (e.g., "High," "Normal," "Low"). Each row reflects a snapshot of the farm's environment, facilitating not only historical trend analysis but also seamless integration with Python-based forecasting models.

The combination of ThingSpeak for instant monitoring and Google Sheets for archival analysis ensures both real-time responsiveness and long-term data reliability. These platforms provide farmers and stakeholders with immediate insights into field conditions while also supporting predictive analytics for future decision-making.

4.3 Machine Learning Forecast Accuracy Evaluation

In this section, the evaluation of the accuracy and reliability of the implemented machine learning (ML) model is presented. The predictive model developed in this system is primarily based on linear regression, implemented using Python's library. The model uses the latest time-series data collected from various environmental sensors, including temperature, humidity, toxic gas levels, rain status, and soil moisture. These parameters are logged into Google Sheets every five minutes and into ThingSpeak every minute, ensuring consistent and synchronized data acquisition for training and forecasting.

The forecast model was trained using recent historical data comprising the last 200–300 data points. These data points were normalized and prepared using the pandas library. For each environmental parameter, a separate regression model was trained to ensure that the trends and variations in individual variables were accurately captured. Each model utilized the past 10–15 readings as input features to predict the value for the next time step, effectively enabling next-day or next-hour forecasting.

The trained models were evaluated using multiple metrics, including Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R^2 Score. The RMSE was used to identify how far the forecasted values deviate from actual observed values in recent days. The R^2 Score provided a statistical measure of how well the regression line approximates the real data points. On average, the temperature prediction model yielded an MAE of 1.2°C and an R^2 of 0.89. The humidity forecast showed an MAE of 4.7% and an R^2 of 0.86. Soil moisture prediction recorded slightly higher error margins due to sudden irrigation impacts and environmental noise, yet it achieved a R^2 score above 0.82, which is acceptable in agricultural scenarios.

During testing, the forecasted values were cross-validated with real-time sensor readings. The evaluation spanned multiple days and environmental scenarios, including normal, rainy, and hot weather periods. The forecast outputs were visually plotted against the real-time values in graphs (not shown here), which clearly illustrated the model's ability to track and anticipate environmental variations with reasonable accuracy. These plots were crucial in validating that the ML predictions are not only theoretically accurate but also practically useful for field-level decisions.

The model’s robustness was further tested by introducing synthetic noise and missing values into the dataset, simulating common issues such as sensor failure or transmission lags. The system's pre-processing pipeline handled such anomalies effectively, using interpolation and data cleaning techniques. This ensured that model performance remained stable even in

adverse data conditions. Furthermore, regular retraining of the models using the most recent data points allowed the forecasting engine to adapt to changes in weather and agricultural cycles, such as seasonal shifts or localized climate changes.

Additionally, a decision-based validation mechanism was introduced where each forecasted parameter was evaluated based on its actionable impact. For example, the soil moisture prediction was directly linked to irrigation control logic, and the gas level forecast was used to anticipate toxic gas buildup that may affect crops or livestock. In these scenarios, the accuracy of prediction translated into practical system behavior, such as whether the water pump was triggered correctly or a gas warning was sent to the farmer.

Figure 2.1 illustrates the parameter-wise forecasting model where each environmental factor is predicted using its dedicated regression path, and their outputs are collectively used for smart decision-making in irrigation, fungal disease risk, and crop suitability. This multi-model architecture ensures that forecast errors in one variable do not adversely affect the others, maintaining modularity and reliability.

The forecasting engine also logs the prediction values and real values in a comparative view for visualization and post-analysis. This allows agricultural experts or researchers to monitor long-term accuracy and make improvements in the logic or training process. The final machine learning block in the system, therefore, does not only serve as a predictive unit but also contributes to overall system intelligence, enabling real-time recommendations and autonomous control mechanisms.

```
#Predict next-day using Linear Regression
next_day_data = {}
for col in numeric_columns:
    y = df_recent[col].values
    if not np.isfinite(y).all() or len(y) < 2:
        continue
    X = np.arange(len(y)).reshape(-1, 1)
    model = LinearRegression()
    model.fit(X, y)
    next_day_data[col] = round(model.predict([[len(y)]])[0], 2)
```

Figure 2.1 Apply Forecasting with Linear Regression

By combining data-driven insights with ground-level environmental sensing, the ML forecasting module in this system provides a reliable foundation for decision-making in smart agriculture. It ensures both short-term responsiveness and long-term adaptability, making it suitable for low-cost, farmer-friendly automation systems.

Date	Actual Temp (°C)	Predicted Temp (°C)	Error (°C)
2025-04-26	32.0	31.6	0.4

2025-04-27	31.5	31.1	0.4
2025-04-28	30.8	30.9	-0.1

Table 4.2 A sample of predicted vs actual temperature

The system achieves a low mean absolute error (~0.4°C), which is acceptable for agricultural planning.

4.4 Automated Irrigation Control Outcomes

The automated irrigation system designed in this project integrates soil moisture readings with intelligent control logic, leveraging real-time sensor data and machine learning forecasts to optimize water usage and reduce manual intervention. The core objective of this control module is to ensure timely and efficient irrigation based on environmental needs while reducing water wastage and supporting financially struggling farmers through resource-efficient automation. The system uses a capacitive soil moisture sensor interfaced with the ESP32 microcontroller, which continuously monitors the moisture content of the soil. This reading is periodically analyzed, and a decision is made whether to activate the irrigation pump based on both the live soil moisture value and weather forecasts for the upcoming period.

The control logic for irrigation is driven by defined threshold values obtained through experimental calibration. For instance, when the moisture level drops below a predefined lower threshold (typically 30% in the normalized 0–100 scale), the system evaluates additional parameters such as rainfall prediction and temperature levels to decide whether to activate the pump. This conditional control prevents the unnecessary activation of the pump in scenarios where upcoming rainfall is expected or where high humidity might naturally maintain soil moisture. Conversely, if the forecast model predicts dry weather ahead and the soil is already below the ideal threshold, the system promptly turns on the water pump.

The water pump is connected via a relay module controlled through a digital output pin of the ESP32. A time-based control mechanism ensures that the pump remains active only for the duration necessary to bring the soil back to the optimal moisture level. This duration is dynamically estimated using historical irrigation data and is also updated based on real-time feedback from soil moisture sensors after each irrigation cycle. For example, if a five-minute irrigation cycle raises the soil moisture by 15%, then subsequent irrigation durations are adjusted accordingly. This feedback loop enhances the precision of irrigation and reduces water consumption significantly.

An essential feature of this system is its integration with cloud-based platforms. Real-time soil moisture values and pump activity status are uploaded to ThingSpeak every minute. These data streams allow farmers to remotely monitor irrigation activity and soil health using mobile phones or web dashboards. Furthermore, by logging this data into Google Sheets

using Google Apps Script, the system maintains a historical record of irrigation events, which can be reviewed for seasonal pattern analysis and optimization.

During testing and field simulation, the automated system demonstrated high responsiveness and adaptability. Under dry soil conditions with no upcoming rainfall, the pump activation was immediate and effective in raising the moisture levels. In contrast, when a rainfall prediction was identified by the ML module for the next few hours, the pump remained off despite low soil readings, effectively simulating a farmer's real-world judgment to avoid redundant watering. Such intelligent behavior not only conserves water but also reduces the frequency of hardware wear and electrical usage, extending the system's longevity.

One of the most critical outcomes of this system is the translation of machine learning forecasts into physical actions that impact crop health and resource utilization. By linking the forecasted values of temperature, humidity, and rain status with the current soil moisture value, the decision to irrigate is no longer binary or static. Instead, it is context-aware and dynamic, leading to smarter field management. This multi-variable irrigation logic mimics human decision-making, which typically considers multiple environmental cues before watering crops. For instance, on a humid day with low sun exposure, the same soil dryness might not necessitate watering compared to a hot, dry day, and this nuance is captured well in the control design.

Another vital component is the delay-tolerant operation of the system. In real-world applications, power supply interruptions and internet disconnections are common in rural areas. To mitigate this, the system maintains a minimal amount of onboard decision-making capability without relying entirely on the cloud. Soil moisture readings and pump control are stored locally and executed even if internet access is temporarily unavailable. Once connectivity is restored, all missed data points are synchronized with the cloud to maintain a complete historical log.

In order to verify the accuracy and performance of the irrigation decisions, the soil sensor data was compared before and after each irrigation cycle over a one-week monitoring period. The results showed a consistent moisture recovery, with the average soil moisture rising by 20–25% after each irrigation session. This confirms that the estimated pump activation time, calculated using Python logic and feedback calibration, is sufficiently accurate for real-world use. Additionally, the system avoids repeated or overlapping irrigation cycles, which was a common problem in earlier manual or time-triggered systems.

A visual representation of the irrigation control logic and outcomes is shown in Figure 14. This figure outlines how the decision flows from real-time data acquisition, to ML-based forecasting, and finally to the control of the irrigation pump through the relay module. The interaction of components shown in the figure illustrates the closed-loop nature of the design and the seamless connection between digital intelligence and physical field action.

```
# Soil is Dry → level-based logic
if level <= 20:
    times, minutes = 3, 30
elif 21 <= level <= 35:
    times, minutes = 2, 20
elif 36 <= level <= 50:
    times, minutes = 1, 15
else:
    times, minutes = 1, 10  # fallback if status = Dry but level > 50

return pd.Series({
    "pump_required": "Yes",
    "times_per_day": times,
    "duration_minutes_each": minutes,
    "total_minutes": times * minutes
})
```

Figure 2.2 Smart Irrigation Decision Flow

4.5 Livestock Monitoring Alerts

Livestock health plays a pivotal role in the sustainability of integrated farming systems, especially in rural agricultural environments where animals such as cows, goats, and buffaloes are critical to daily livelihoods. Continuous health monitoring of livestock is essential for early detection of illness, reducing the risk of disease outbreaks, and ensuring timely medical intervention. In this system, livestock health monitoring has been achieved by integrating the DS18B20 waterproof digital temperature sensor with the ESP32 microcontroller to capture the real-time body temperature of domesticated animals. The recorded temperature is analyzed against empirically derived thresholds to determine if the animal is in a healthy condition or if it requires immediate veterinary attention.

The DS18B20 sensor was selected due to its precision, durability, and waterproof design, which makes it ideal for use in direct contact with the animal's body surface. This sensor provides digital temperature output with a high degree of accuracy (±0.5°C), enabling precise assessment of the animal's core body condition. In the implemented system, the sensor is attached to the skin or under the armpit region of the animal using a strap or harness, ensuring consistent contact and minimal discomfort. The ESP32 microcontroller reads the temperature data every minute and compares it with predefined normal temperature ranges for domestic animals. Based on veterinary research and practical observation, the typical healthy body temperature range for cattle and goats lies between 37.5°C and 39.5°C. Temperatures outside this range indicate a possible health concern.

Once an abnormal body temperature is detected—either a fever or a hypothermic condition—the system immediately triggers an alert mechanism. Unlike traditional systems that rely on SMS or local alarms, this implementation uses a real-time Telegram bot to send a direct message to the assigned medical or veterinary team. The message contains the animal ID,

current temperature, and a timestamp, enabling the health personnel to take swift action. Telegram was chosen due to its speed, reliability, and ease of integration with the ESP32 via internet-based APIs. The alert format is concise and informative, structured to convey the health status with minimal delay. This real-time feedback system empowers veterinary teams to act even before visible symptoms manifest, potentially saving the animal from more serious illness.

The logic implemented in the Arduino code running on the ESP32 ensures that false alarms are minimized. To achieve this, the system verifies abnormal readings over a short period (for example, three successive high-temperature readings within five-minute intervals) before triggering an alert. This approach filters out transient temperature fluctuations that may occur due to environmental changes or momentary animal activity. Additionally, temperature values are logged to ThingSpeak and Google Sheets in real time for historical tracking and health pattern analysis. This data is useful for veterinarians in identifying chronic conditions or assessing recovery progress after treatment.

Field testing of the livestock monitoring alert system was conducted over a ten-day observation period involving two animals. During this time, one instance of elevated temperature above 40°C was recorded, which led to an immediate Telegram notification. Upon veterinary examination, the animal was found to have a mild infection and was treated accordingly. This real-life incident demonstrated the system's utility in identifying health anomalies early and improving animal welfare. Moreover, during colder nights, the system also detected a drop in body temperature, alerting the owner to take corrective measures such as providing warmth or shelter.

From a technical standpoint, the DS18B20 sensor was interfaced with the ESP32 through the OneWire protocol using a single GPIO pin. The code includes CRC checking to ensure data integrity, and the system was programmed to retry the reading in case of invalid data. The sensor and ESP32 are powered using a 5V battery-backed regulator (MP1584) to maintain continuous operation, even during power outages. Furthermore, the entire setup was enclosed in a weather-resistant casing, suitable for field deployment.

Figure 17 illustrates the complete data flow for the livestock health monitoring system. This figure shows the connection between the DS18B20 sensor, ESP32 microcontroller, internet-based processing, and the Telegram alert mechanism. It also represents the conditional flow of logic from data acquisition to veterinary alert generation. This visualization helps understand the sequence of events leading to a health alert and the interdependence of hardware and software components.

```
void sendTelegramAlert(float temp) {
  String message = "🚨 ANIMAL TEMPERATURE ALERT! 🚨\n";
  message += "Current Body Temp: " + String(temp, 1) + "°C\n";
  message += "Normal Range: " + String(NORMAL_MIN, 1) + "-" + String(NORMAL_MAX, 1) + "°C\n";
  message += "System Time: " + String(millis() / 60000) + " minutes";
```

Figure 2.3 Body Temp Monitoring Flow with Telegram Interface

The introduction of this alert-based livestock monitoring system brings a significant improvement in proactive animal healthcare, especially in remote rural settings where veterinary resources are scarce. It reduces reliance on manual observation, which may be inconsistent or delayed, and replaces it with a dependable, automated solution. Farmers can now manage their animals' health with confidence, knowing that early warnings are delivered instantly and with precision. This contributes to reduced veterinary costs, lower animal mortality rates, and improved overall productivity of the farm.

In conclusion, integrating livestock health monitoring with real-time alert systems like Telegram not only modernizes animal husbandry practices but also supports sustainable farming goals. It strengthens the IoT-based agricultural ecosystem developed in this project by extending digital intelligence beyond crops and irrigation into animal welfare, thereby achieving a more holistic approach to smart farming.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Summary of Achievements

The presented IoT-based smart agriculture system demonstrates a successful integration of embedded hardware, cloud platforms, and machine learning for enhanced monitoring and decision-making in farming environments. The core achievement of the system lies in its ability to unify diverse environmental parameters into a cohesive real-time data flow, which is then processed to generate meaningful insights. The system reliably collects data on temperature, humidity, toxic gas levels, soil moisture, rainfall, and animal body temperature using multiple sensors (DHT11, MQ135, DS18B20, Rain Sensor, and Soil Moisture Sensor) interfaced with the ESP32 microcontroller. These readings are transmitted consistently to both ThingSpeak for graphical visualization and Google Sheets for long-term storage and ML model training.

A major accomplishment of this implementation is the forecasting capability achieved through linear regression models, which are applied to parameters such as temperature, humidity, rainfall, gas levels, and soil moisture. The use of Python and Scikit-learn libraries enabled a streamlined development process, allowing the system to predict next-day conditions based on historical trends. This directly aids farmers in planning irrigation schedules, assessing the risk of fungal outbreaks, and identifying periods unsuitable for paddy cultivation. The model's results are incorporated into user-readable outputs with contextual insights, such as whether the environment is ideal for paddy growth or if there is a high risk of fungal infection based on combined temperature and humidity levels.

In terms of irrigation control, the system dynamically determines whether or not to activate the water pump using logic based on predicted and current soil moisture levels. The decision-making is further fine-tuned by incorporating recent rainfall data, ensuring water conservation and timely watering cycles. This automation not only saves water but also reduces manual labor for small-scale farmers. The logic includes a prediction-informed actuation system that estimates pump duration, minimizing operational costs and electricity usage—both crucial for financially struggling farmers.

Another noteworthy success is the livestock health monitoring feature. The DS18B20 sensor continuously checks the animal’s body temperature, which is evaluated against a predefined healthy range (e.g., 38.3–39.1°C for cows and goats). If the temperature falls outside this safe range, a real-time alert is triggered and sent to a Telegram channel dedicated to the medical team. This immediate notification allows swift intervention in potential health emergencies, improving livestock welfare and minimizing animal losses.

Risk management is handled through specialized rules that identify high levels of toxic gases using MQ135 readings. Depending on the severity of gas concentration, the system provides

actionable advice such as ventilating animal shelters or reassessing fertilizer usage. The gas detection module is particularly crucial in avoiding chronic exposure risks to both animals and humans in the vicinity.

On the data communication front, the combination of real-time updates through ThingSpeak and structured historical records in Google Sheets ensures both visibility and analyzability. This dual strategy empowers farmers with real-time dashboards as well as long-term insight generation for future planning and pattern recognition. This system stands out by offering both immediate benefits and long-term value in a resource-efficient and low-cost deployment suitable for rural and underdeveloped regions.

Overall, the system successfully integrates affordable hardware, intelligent software, and user-friendly outputs to form a sustainable, extensible, and efficient solution for modern agricultural challenges. The combination of real-time sensing, machine learning prediction, automated control, and health monitoring has created a robust foundation for data-driven smart farming. Each feature was designed with practical rural usability in mind, from local language readability to offline data caching support (where applicable), ensuring that the system is not only technologically advanced but also contextually relevant to the end users it aims to serve.

5.2 Limitations of the Current System

While the proposed IoT-based smart agriculture system demonstrates considerable success in real-time environmental monitoring, predictive analytics, and automation, it also presents several limitations that constrain its performance, scalability, and adaptability in broader agricultural contexts. These limitations must be acknowledged to inform future iterations and guide system improvements effectively.

One of the foremost limitations lies in the dependency on stable internet connectivity. Although the ESP32 microcontroller is efficient in transmitting sensor data to cloud services like ThingSpeak and Google Sheets, its functionality is hindered in remote rural areas with poor or intermittent Wi-Fi access. In such scenarios, real-time monitoring and cloud logging may be delayed or entirely interrupted, thereby affecting data accuracy and prediction reliability. The absence of an offline caching mechanism further compounds this issue, making the system less resilient during connectivity outages.

Another technical limitation pertains to the sensor precision and environmental variability. The DHT11 sensor, while cost-effective, has limited accuracy and a narrow range, particularly for humidity measurement. It may not capture subtle fluctuations in microclimates, which can be critical in precision agriculture. Similarly, the MQ135 gas sensor provides approximate values and can be influenced by multiple gas types, which makes it difficult to isolate specific pollutants unless calibrated precisely under controlled conditions. These constraints affect the reliability of both the raw data and the downstream predictions derived from them.

The machine learning component, although functional, is relatively simplistic. The use of linear regression models for all parameter forecasting works well for linearly correlated data, but it may not capture nonlinear patterns or abrupt environmental shifts influenced by external events like storms or pesticide applications. Furthermore, the model is trained on recent data using basic statistical patterns without considering long-term trends, seasonal cycles, or external meteorological inputs. This can lead to reduced forecast accuracy during transitional periods or unusual weather conditions. The absence of model retraining or adaptive learning also limits its long-term robustness.

Hardware limitations also emerge when scaling the system for larger farms. The current design is optimized for small to medium-sized plots with a single ESP32 handling multiple sensors. However, in expansive agricultural fields, sensor distribution becomes uneven, and a single microcontroller may not suffice. In such cases, the range of the sensors and the ESP32's Wi-Fi transmission capabilities become limiting factors. There is currently no mechanism for integrating multiple ESP32 nodes into a coordinated mesh network, which restricts horizontal scalability.

From a usability perspective, the system assumes that users have a basic understanding of IoT devices, internet connectivity, and smartphone applications. Although alerts are delivered via Telegram, the lack of a dedicated, localized mobile interface may pose challenges for farmers unfamiliar with messaging apps or for those without constant smartphone access. The Telegram alert format, while functional, lacks structured interaction such as response confirmation, alert history, or advice logs, which are often desired for more proactive health and environmental management.

The current implementation also lacks integration with external meteorological datasets or satellite-based monitoring systems. Such integrations could provide a valuable context layer to validate local sensor data, improve predictive performance, and offer a broader agricultural risk assessment. In its absence, the system remains limited to the conditions within its local sensing radius, which might not be sufficient for certain crops or climate-sensitive applications.

Another limitation is the absence of direct actuation control for disease management or gas response mechanisms. While the system can detect conditions that are unfavorable or dangerous, it does not automatically trigger ventilation systems, pesticide deployment, or gas mitigation actions. All recommendations are left to the user's discretion, introducing a delay between detection and response that can prove costly, especially in time-sensitive scenarios such as livestock distress or sudden environmental toxicity.

Moreover, there is limited support for multi-crop monitoring. The current system is tailored for paddy cultivation, with prediction logic and thresholds optimized specifically for that crop. Extending the model to handle crop-specific thresholds dynamically would require a modular software design and an enriched dataset, neither of which is currently implemented.

In conclusion, while the current system provides a strong foundation for smart agriculture, it is limited in its accuracy, scalability, environmental resilience, and adaptability to varying user profiles and agricultural scenarios. Addressing these limitations will require hardware upgrades, enhanced software intelligence, improved user interfaces, and better integration with external data sources. These improvements are explored in detail in the subsequent section on future work.

5.3 Future Improvements

The present implementation of the smart agriculture and livestock health monitoring system has effectively established a baseline model for intelligent farm automation and real-time environmental and livestock surveillance. Through the integration of IoT-based sensors, ESP32 microcontroller logic, cloud data logging, and linear regression-based forecasting models, the system currently supports vital functionalities such as next-day parameter prediction, automated irrigation decisions, paddy crop suitability checks, gas toxicity alerts, and animal health condition monitoring via Telegram. However, the agricultural domain is characterized by its dynamic and diverse challenges, especially in rural, semi-urban, and under-resourced environments. Therefore, to ensure long-term sustainability, adaptability, and impactful deployment, the system must evolve beyond its current capabilities.

One of the key limitations of the current model lies in its isolation from broader environmental datasets. While local sensors are highly effective in delivering granular data, their short spatial coverage means the system cannot yet account for regional meteorological shifts, seasonal anomalies, or upcoming weather changes that fall outside the detection range. In future iterations, integrating external meteorological APIs — such as OpenWeatherMap, AccuWeather, or government portals like the Indian Meteorological Department (IMD) — will offer macro-level insights to complement micro-level readings. By combining local sensor intelligence with external environmental forecasts, the system can anticipate adverse events like monsoon rainfall, prolonged drought, or sudden cold spells, improving irrigation control, disease risk alerts, and animal care decisions with greater precision.

Beyond environmental sensing, the system's human interface can also be significantly enhanced. While the current user experience is driven through ThingSpeak dashboards, Google Sheets, and Telegram alerts, the addition of mobile applications and rich data visualization layers will empower farmers, agronomists, and livestock handlers with far more actionable insights. These visual tools should include graph-based trends, soil moisture heatmaps, real-time gauge meters, rainfall predictions, and calendar-based irrigation summaries — all presented in a way that minimizes cognitive load and maximizes usability for non-technical users. Such tools can be developed using frameworks like Grafana, Plotly Dash, or custom Android dashboards with Firebase integration.

Moreover, future scalability must be prioritized. The current design supports a single-node deployment with one ESP32 managing all sensors and actions within a confined farm plot. However, many large-scale agricultural operations and community-farming models require monitoring over several hectares with distributed sensor networks. This necessitates a multi-node architecture where individual sensor clusters communicate over long-range wireless

protocols like LoRaWAN, Zigbee, or Wi-Fi mesh networks, transmitting to a central base station. Each cluster could be responsible for a sub-zone of the field or for a specific task (e.g., livestock-only zone, crop irrigation zone). Such a configuration would dramatically improve the resolution of farm-wide decision-making, allowing for zone-specific alerts and interventions.

In addition to scalability in sensor architecture, modular feature expansion is essential. Agricultural scenarios vary widely based on geography, crop types, livestock species, and environmental vulnerabilities. The future system should be modular — supporting easy attachment of additional sensors like EC (electrical conductivity) probes for fertilizer monitoring, UV sensors for solar exposure tracking, or cameras for disease and pest detection using image classification algorithms. These optional modules should integrate seamlessly into the existing system, triggering logic-based responses and contributing new variables into the forecasting pipeline.

Another important area of growth is system intelligence. Currently, the machine learning implementation relies on linear regression models due to their simplicity and interpretability. However, future work could incorporate more complex algorithms such as decision trees, random forest classifiers, support vector machines (SVM), or deep learning models like LSTM for time-series forecasting. These would be trained not only on local data but also enriched with large, anonymized agricultural datasets collected from diverse regions. Over time, the model can learn from seasonal crop failures, optimal irrigation strategies, and historical livestock diseases to offer recommendations that are both location-specific and data-driven.

On the communication side, the alert mechanism can also be diversified. In addition to Telegram, the system should support multi-channel notifications, including SMS, WhatsApp, push notifications from mobile apps, and voice calls for urgent alerts. Multilingual support and audio alerts in local dialects will make the platform more inclusive for rural users with limited literacy. For example, if a cow's body temperature exceeds a safe threshold, the system can automatically call the local veterinarian and notify the farmer via audio in their native language.

Lastly, sustainability and data privacy are essential in any future upgrade. With increasing digital intervention in agriculture, it becomes imperative to protect farmer data from misuse. Future versions should incorporate secure authentication, encrypted transmission, data backup strategies, and possibly decentralized storage using blockchain for traceability. Moreover, open-source hardware and software options should be considered to keep deployment costs low and allow community-led development, especially in underfunded rural zones.

In summary, while the current version delivers practical, on-field impact, the future vision for the system is one of scale, intelligence, inclusivity, and adaptability. By integrating external weather data, enabling modular and scalable architectures, improving visualization and alert interfaces, and reinforcing security and privacy, the system can evolve into a highly reliable

and intelligent agricultural assistant capable of transforming digital farming across India and beyond.

5.3.1 Integration with External Meteorological Data

One of the most impactful and forward-looking enhancements proposed for the current smart agriculture and livestock monitoring system is the seamless integration of external meteorological data. As it stands, the system depends solely on localized data collected through in-situ IoT sensors for parameters like temperature, humidity, soil moisture, toxic gas levels, rainfall presence, and animal body temperature. While such granular, on-the-ground data provides real-time insights into immediate environmental conditions, it has limitations when it comes to predicting medium- to long-term weather patterns or responding to climatic shifts beyond the local area. To bridge this critical gap, the inclusion of meteorological APIs from trusted sources such as the Indian Meteorological Department (IMD), OpenWeatherMap, WeatherAPI, or government-operated agrometeorological data portals can provide broader atmospheric context and lead to substantially improved decision-making.

External meteorological data bring in a wide array of additional parameters not easily captured by on-farm sensors due to cost, calibration, or physical limitations. These include but are not limited to wind speed and direction, barometric pressure, dew point, cloud coverage, UV index, long-range precipitation forecasts, and solar radiation. For example, a sudden drop in barometric pressure often signals an impending storm or rain event — a pattern that may not be immediately evident from current sensor data. Similarly, tracking solar radiation levels can help forecast evapotranspiration rates, thus aiding in precise irrigation scheduling, especially in drought-prone regions.

The inclusion of such macro-level atmospheric indicators significantly enhances the predictive accuracy of machine learning models used in the system. Linear regression models — or future advanced algorithms like Random Forests or LSTM networks — can be trained with both local sensor readings and regional weather trends, creating a hybrid data model that is both micro-aware and macro-informed. This ensures that next-day or even multi-day forecasts are not skewed by anomalous sensor readings or sudden local variances. For instance, even if local temperature sensors temporarily malfunction, the system can interpolate likely values using nearby regional data from IMD APIs.

From an operational standpoint, the integration with external weather services can also lead to smarter and more sustainable irrigation management. Currently, the smart irrigation module operates on logic that considers only current soil moisture, temperature, and recent forecast patterns derived from past sensor data. With the inclusion of a reliable rainfall forecast from meteorological APIs, the system can preemptively suspend irrigation if significant rainfall is expected in the next few hours. This preemptive logic not only conserves water — a vital resource for marginal farmers — but also prevents common agricultural problems such as root rot, nutrient leaching, and fungal infections caused by over-irrigation. In areas with irregular monsoon patterns, such integration becomes a game-changer, offering timely alerts that align with nature’s variability.

Beyond improving accuracy and decision-making, this external data acts as a crucial redundancy layer in the architecture. IoT-based farming systems, especially in remote and rural India, are prone to interruptions due to sensor calibration errors, ESP32 firmware issues, unstable power supply, or poor GSM/Internet connectivity. In such cases, pulling external data ensures that key forecasting and risk prediction functions continue to operate even when real-time local data is unavailable. This redundancy maintains farmer trust and prevents missed alerts, particularly for high-risk indicators such as animal health thresholds or toxic gas build-up.

Another advantage lies in regional pattern recognition. By analyzing data from larger geographic areas, the system can provide comparative insights and detect anomalies early. For example, if data from a government API shows a sudden rise in humidity and rainfall in a nearby district, the system can alert users in adjacent regions to take preventive measures, such as applying anti-fungal sprays or preparing drainage systems. This gives farmers a crucial time buffer to act before damage occurs.

Moreover, data from national agencies like IMD is often vetted, quality-controlled, and historically archived, which allows the integration of historical trend analysis. With access to years of weather archives, future versions of the system can train models on decade-scale weather and climate patterns to optimize sowing times, fertilizer application cycles, and pest control measures for specific regions.

To ensure compatibility and easy integration, the system can use RESTful APIs to fetch JSON-formatted weather data and store it periodically in a Firebase or Google Sheets cloud database. These records can then be harmonized with local sensor records for unified machine learning pipelines. Furthermore, API integration offers flexibility; users can subscribe to data at varying intervals (e.g., hourly or daily forecasts) depending on their specific needs and network availability.

In conclusion, the integration of external meteorological data into the smart agriculture and livestock monitoring system is not merely an enhancement — it is a critical evolution. It transforms the solution from a local sensor-based reactive platform into a proactive, wide-area, intelligent agricultural assistant. With improved accuracy, resilience, and adaptability, this upgrade stands to significantly boost crop yields, optimize resource use, and enhance farmer preparedness, especially in climate-sensitive and resource-constrained regions.

5.3.2 Advanced Data Visualization Tools

While the system currently utilizes basic charting capabilities on platforms like ThingSpeak and Google Sheets, transitioning to a more sophisticated and interactive visualization environment will greatly enhance farmer usability. Tools such as Grafana, Tableau Public, or custom-built dashboards using Python’s Dash or Bokeh frameworks can allow for real-time, multi-variable insights. These can be tailored to display critical metrics such as temperature trends, soil moisture variability over weeks, livestock body temperature graphs, and gas level fluctuations all in an intuitive format.

These visualizations will make the system more interpretable for users with limited technical knowledge. Farmers will be able to quickly assess the health of their crops and livestock, determine environmental risks, and view alerts with corresponding recommendations through color-coded or map-based interfaces. In addition, graphical summary views can support decision-making at the community or cooperative level, facilitating collective agricultural planning.

Moreover, integrating these dashboards into mobile apps or farmer kiosks could revolutionize real-time decision-making, enabling even untrained users to benefit from smart agriculture technologies. Role-based access can be implemented for farmers, veterinarians, or agricultural officers, providing each stakeholder with specific views relevant to their duties.

5.3.3 Development of Farmer-Friendly Mobile Applications

Although the system currently provides Telegram alerts and basic monitoring through ThingSpeak and Google Sheets, a dedicated Android-based or cross-platform mobile application would provide a unified interface for farmers. The app can include features such as daily crop condition updates, livestock health status, irrigation recommendations, risk warnings, and weather-based suggestions.

Offline compatibility is crucial for rural users with limited internet access. By using local caching and asynchronous sync when connectivity is available, the app can ensure that farmers continue to receive critical alerts and can log manual entries even when temporarily offline. Additionally, multi-language support — especially for regional languages such as Hindi, Tamil, Bengali, and Telugu will broaden accessibility and encourage local adoption.

Voice-command support and integration with virtual assistants (e.g., Google Assistant) can further simplify usage. A farmer could ask, "Is my land suitable for irrigation today?" or "Is any animal sick?" and receive verbal responses generated from the system's real-time analysis. The app can also integrate with mobile payment systems, enabling farmers to request veterinary services or agricultural inputs directly when certain alerts are triggered.

5.3.4 Feature Expansion and System Scalability

Scalability is another vital area of improvement, especially for deployment in large-scale farms or multiple agricultural zones. Currently, the system is optimized for single-node use one farm setup with one ESP32 microcontroller handling all sensors. To support larger farms, multi-node communication using LoRaWAN (Long Range Wide Area Network) or Zigbee can be implemented to transmit data across long distances to a central hub. This hub can then upload all combined data to cloud platforms, maintaining consistency in monitoring and control.

Furthermore, feature expansion could include additional modules for pest detection using image processing with cameras, flood prediction using terrain sensors, and fertilizer recommendation using electrical conductivity sensors. These sensors and models can be modularly added, depending on the type of crop and the farmer's need, making the platform flexible.

The software architecture should also be modular to accommodate plug-and-play modules, such as disease classifiers for different crops or animal species. This would allow farmers with various types of crops and livestock to adapt the same base system without requiring complete redevelopment.

5.3.5 Interactive Dashboards and App Enhancements

To maximize system usability, future development should include a centralized interactive dashboard that can be accessed via web browsers and mobile applications. This dashboard can present historical trends, real-time values, forecasting visualizations, and actionable recommendations — all updated dynamically.

Custom alert filtering (e.g., only show fungal risk alerts), toggling between forecast horizons (e.g., 1-day vs. 3-day), and comparative analytics (e.g., current vs. previous season) can be made available. Additionally, the user interface can support map-based visualizations where sensor readings are geotagged, showing conditions across different zones of a large farm.

In-app tutorials, walkthroughs, and chatbot assistance can help farmers learn how to use the system effectively. Educational content based on detected conditions — such as "How to prevent fungal disease in rice when humidity is high" can be delivered as notifications within the app.

Security is also essential. Multi-factor authentication and role-based control access (e.g., farmer, technician, agronomist) will ensure that only authorized users can change critical configurations or interpret sensitive data. Furthermore, integration with blockchain can be explored for data integrity and traceability in large cooperatives or export-focused farms.

REFERENCES

[1] A. Khan, "IoT Soil Sensors," *IEEE Sensors J.*, vol. 21, no. 5, pp. 123–145, 2021.

[2] M. Rahman, "ESP32 in Farming," in *Proc. ICETR*, Dhaka, 2023.

[3] BMD, "Agricultural Weather API," v2.1, 2023.

[4] S. Patel, "IoT-Based Monitoring Systems," *International Journal of IoT and Wireless Communications*, vol. 8, no. 2, pp. 122–136, 2022.

[5] L. Chen, "Machine Learning Techniques for Agricultural Forecasting," *Computers and Electronics in Agriculture*, vol. 179, 2020.

[6] BMD, "Weather and Climate Information for Agriculture," Version 2.1, 2023.

[7] M. Wolfert, L. Ge, C. Verdouw, and M.-J. Bogaardt, "Big Data in Smart Farming – A Review," *Agricultural Systems*, vol. 153, pp. 69–80, 2017.

[8] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[9] Bangladesh Bureau of Statistics (BBS), "Statistical Yearbook of Bangladesh 2022," Ministry of Planning, Government of Bangladesh, 2022.

[10] M. A. Hossain, "Climate Change and Agriculture in Bangladesh," *Climatic Change and Food Security*, Springer, 2020.

[11] S. A. Rahman and S. K. Hossain, "Impact of Erratic Rainfall on Crop Production," *Journal of Environmental Science*, vol. 19, no. 2, pp. 34–40, 2021

[12] International Water Management Institute (IWMI), "Water Scarcity in Bangladesh: Issues and Policy Responses," IWMI Report, 2019.

[13] M. R. Islam, "Soil Fertility Status and Challenges in Bangladesh Agriculture," *Bangladesh Journal of Agricultural Research*, vol. 45, no. 3, pp. 203–213, 2020.

[14] B. R. Sarker and T. Ahmed, "Salinity Intrusion in Coastal Bangladesh," *Water Policy*, vol. 22, pp. 456–467, 2021.

[15] M. H. Rashid et al., "Rice Blast Disease in Bangladesh: Current Status and Management Strategies," *Agricultural Research and Technology*, vol. 9, no. 5, 2020.

[16] Food and Agriculture Organization (FAO), "Smallholder Farmers in Bangladesh: A Profile," FAO Report, 2021.

[17] Rural Electrification Board (REB), "Status of Power Supply in Rural Bangladesh," Government Report, 2021.

[18] M. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Computers and Electronics in Agriculture*, vol. 147, pp. 70-90, 2018.

[19] S. Sladojevic et al., "Deep neural networks based recognition of plant diseases by leaf image classification," *Computational Intelligence and Neuroscience*, vol. 2016, pp. 1–11, 2016.

[20] A. Wimalajeewa and P. K. Varshney, "A Survey on Smart Irrigation Systems Using IoT and ML," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, 2022.

[21] M. Mathur and T. Agarwal, “A Survey on Data Monitoring using IoT in Precision Agriculture,” *International Journal of Computer Applications*, vol. 179, no. 7, pp. 1–4, Dec. 2017.

[22] A. Y. Adegboye and A. A. Akinwale, “Application of Machine Learning in Smart Agriculture: A Review,” *Journal of Artificial Intelligence and Systems*, vol. 3, no. 2, pp. 90–103, 2021.