AI BASED CROP MONITORING SYSTEM

By

HARINI T A	**[713521EE010]**
HARSHINI K S	**[713521EE011]**
KATHIRAVAN B	**[713521EE015]**
KARAN S	**[713521EE509]**

A PROJECT REPORT

Submitted to the

FACULTY OF ELECTRICAL ENGINEERING

In partial fulfilment for award of
the degree
Of

BACHELOR OF ENGINEERING

SNS COLLEGE OF TECHNOLOGY, COIMBATORE-35

(AN AUTONOMOUS INSTITUTION)

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

DECEMBER-2024

AI BASED CROP MONITORING SYSTEM

By

HARINI T A	**[713521EE010]**
HARSHINI K S	**[713521EE011]**
KATHIRAVAN B	**[713521EE015]**
KARAN S	**[713521EE509]**

A PROJECT REPORT

Submitted to the

FACULTY OF ELECTRICAL ENGINEERING

In partial fulfilment for award of

the degree

Of

BACHELOR OF ENGINEERING

SNS COLLEGE OF TECHNOLOGY, COIMBATORE-35

(AN AUTONOMOUS INSTITUTION)

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

DECEMBER-2024

BONAFIDE CERTIFICATE

Certified that this report titled **"AI BASED CROP MONITORING SYSTEM"** is a Bonafide work of **HARINI T A [713521EE010], HARSHINI K S [713521EE011], KATHIRAVAN B [713521EE015]** and **KARAN S [713521EE509]** who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	**SIGNATURE**
Dr.C.RAMAKRISHNAN	**Dr.C.RAMAKRISHNAN**
PROFESSOR	**PROFESSOR**
HEAD OF THE DEPARTMENT	**HEAD OF THE DEPARTMENT**
Department of Electrical and	Department of Electrical and
Electronics Engineering	Electronics Engineering
SNS College of Technology	SNS College of Technology
Coimbatore – 641035	Coimbatore – 641035

Submitted for the VIVA–VOCE examination held at SNS COLLEGE OF TECHNOLOGY, held on …..........................

Internal Examiner **External Examiner**

ACKNOWLEDGEMENT

First of all, we extend our heartfelt gratitude to the Management of **SNS College of Technology** for providing us with all sorts of support in completion of the Project. We record our in debtedness to our Director **Dr.V.P.ARUNACHALAM** and our Principal **Dr.S. CHENTHUR PANDIAN** for their constant support and sustained encouragement for the successful completion of the Project.

Our sincere thanks to **Dr.C.RAMAKRISHNAN,** Professor and Head of the Department of Electrical and Electronics Engineering who has encouraged and given us valuable advice for the completion of this Project in all possible ways.

We whole heartedly place on record our deep sense of gratitude for the valuable motivation and support provide to us by Project Coordinator, **Mr. R. SATHEESH KUMAR** Assistant Professor, Department of Electrical and Electronics Engineering, for making us think innovatively and complete this project.

We take immense pleasure in expressing our humble note of gratitude to our Project Guide **Dr.C.RAMAKRISHNAN** Professor, Professor and Head of the Department of Electrical and Electronics Engineering , for his remarkable guidance and valuable suggestions for completion of the project. We also take this opportunity in thanking for all the way guiding us and supporting us in the industry to complete the project successfully.

We extend our thanks to the Faculty members of our department and also the Technical staff for their help during the project. We also thank our parents and all our friends who have directly or indirectly helped us in this work.

ABSTRACT

Agriculture is a major source of income and employment in India. The most prevalent problem faced by Indian farmers is that they do not select the appropriate crop for their land and do not use the appropriate fertilizer. They will experience a significant drop in production as a result of this. Precision agriculture has been used to solve the farmers' difficulty. Precision agriculture is a modern farming strategy that employs research data on soil properties, soil types, and crop yield statistics to recommend the best crop to farmers as well as fertilizer recommendations based on site-specific features. This decreases the number of times a crop is chosen incorrectly and increases productivity. The problem is solved by proposing a recommendation system through ML models with majority voting technique using Random Forest, Naive Bayes, Support Vector Machine (SVM), Logistic Regression and Random Forest, as learners to recommend a crop for the site-specific parameters with high accuracy and efficiency. In addition to that we are performing real time testing using IOT system. The fertilizer recommendation system is purely python logic based. In this we compare the data (optimum nutrients for growing the crop) with the user's entered data. Then nutrient having maximum difference is made as HIGH or LOW.

.

TABLE OF CONTENT

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	iv
	LIST OF FIGUERS	vi
	LIST OF TABLES	viii
1	**INTRODUCTION**	
	1.1 PROJECT OVERVIEW	1
2	**LITERATURE SURVEY**	
	2.1 INTRODUCTION	2
	2.3 SUMMARY	5
3	**EXISTING SYSTEM**	
	3.1 INTRODUCTION	6
	3.2 TRADITIONAL IRRIGATION	8
	3.3 SENSOR BASED IRRIGATION	8
	3.4 ADVANCED DECISION MAKING(AI)	9
	3.5 LIMITATION OF EXISTING SYSTEM	10
		11
4	**PROPOSED SYSTEM**	
	4.1 THEORETICAL BACKGROUND	13
	4.2 FUNCTIONAL REQUIRMENT	15
	4.3 SYSTEM DESIGN	22
5	**TESTING**	
	5.1 TESTING METHODOLOGIES	36
	5.2 UNIT TESTING	37
	5.3 QUALITY ASSURANCE	37
	5.4 SYSTEM TESTING	38
	5.5 RESULT	39
6	**CONCLUSION AND FUTURE SCOPE**	45

REFERENCES 47

APPENDIX 48

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
Figure 4.1	WATERFALL MODEL	22
Figure 4.2	SYSTEM ARCHITECTURE	23
Figure 4.3	SEQUENCE DIAGRAM	24
Figure 4.4	DFD LEVEL 0	25
Figure 4.5	DFD LEVEL 1	25
Figure 4.6	UML DIAGRAM	26
Figure 5.1	WELCOME PAGE	36
Figure 5.2	CROP RECOMMENDATION	36
Figure 5.3	FERTILIZER RECOMMENDATION	37
Figure 5.4	CROP DISEASE PREDICTION	37
Figure 5.5	DISEASE SUGGESTION	38
Figure 5.6	CROP RECOMMENDATION(OUTPUT)	38
Figure 5.7	DISEASE DESCRIPTION	39

LIST OF TABLES

TABLE NO.	TITLE	PAGE NO.
Table 4.1	YIELD DATASET	27
Table 4.2	TEMPERATURE RAINFALL AND NUTRIENTS DATASET	27
Table 4.3	SOIL NUTRIENTS DISTRIBUTION AS PER CROP	28

CHAPTER 1

INTRODUCTION

A farmer's decision about which crop to grow is generally clouded by his intuition and other irrelevant factors like making instant profits, lack of awareness about market demand, overestimating a soil's potential to support a particular crop, and so on. A very misguided decision on the part of the farmer could place a significant strain on his family's financial condition. Perhaps this could be one of the many reasons contributing to the countless suicide cases of farmers that we hear from media on a daily basis. In a country like India, where agriculture and related sector contributes to approximately 20.4 per cent of its Gross Value Added (GVA) , such an erroneous judgment would have negative implications on not just the farmer's family, but the entire economy of a region. For this reason, we have identified a farmer's dilemma about which crop to grow during a particular season, as a very grave one. The need of the hour is to design a system that could provide predictive insights to the Indian farmers, thereby helping them make an informed decision about which crop to grow. With this in mind, we propose a system, an intelligent system that would consider environmental parameters (temperature, rainfall, geographical location in terms of state) and soil characteristics (N, P, K, pH value, soil type and nutrients concentration) before recommending the most suitable crop to the user. In addition to that a fertilizer suggestion is also made which is based on the optimum nutrients of the crops grown.

1.1 PROJECT OVERVIEW

The project aims to create a smart solution for managing irrigation and monitoring crop health.

Key Features:

- Automated control of water supply based on soil moisture levels.
- Integration of weather forecasts to avoid over-irrigation during rainy days.

CHAPTER 2
LITERATURE SURVEY

S. R. Nandurkar is the Author of the paper titled "IoT-Based Smart Crop-Field Monitoring and Automation Irrigation System" published in 2014 IEEE International Conference on Internet of Things (iThings) This paper focuses on automating irrigation systems using IoT-based smart sensors and Raspberry Pi. Soil moisture, temperature, and humidity sensors are deployed in the field, and data is collected in real time. The system processes this data to decide when irrigation is needed presents an innovative approach to modernizing agriculture using IoT technology. The system incorporates soil moisture and temperature sensors to monitor real-time field conditions, ensuring crops receive optimal care. A Raspberry Pi serves as the central controller, processing sensor data and automating irrigation by activating a water pump when the soil moisture drops below a predefined threshold. Once moisture levels are restored, the pump is turned off, conserving water and preventing over-irrigation. The collected data is stored and analyzed using cloud-based solutions, enabling precision agriculture practices. This system significantly reduces water wastage, enhances crop yields, and minimizes manual labor, providing a sustainable solution for resource-efficient farming. By integrating IoT with agriculture, the paper highlights the potential of technology to address challenges like water scarcity and labor efficiency, ultimately improving productivity in farming.

S. S. Patil is the Author of the paper titled "Intelligent Crop Monitoring and Protection System in Agricultural Field" published in 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC) The paper introduces a system for real-time crop monitoring and pest detection in agricultural fields. It uses a combination of sensors and image processing to monitor crop health and protect against pests. IoT-based system designed to enhance agricultural productivity through real-time monitoring and automated protection mechanisms. The system utilizes a network of sensors to continuously assess soil moisture levels, enabling precise irrigation management that conserves water and optimizes crop growth. Additionally, it incorporates protective measures to safeguard crops from potential for manual intervention and minimizing labor costs. By integrating IoT technology, the system provides farmers with timely data and control capabilities, facilitating informed decision-making and promoting sustainable agricultural practices This system provides an integrated approach to crop health monitoring and protection, enhancing both productivity and resource management

A Kumar Patil is the Author of the paper titled "IoT-Based Intelligent Irrigation and Crop Monitoring System" published in International Journal of Agricultural Technology and Innovation, 2021 This study presents a smart irrigation system that uses IoT sensors to monitor environmental conditions such as soil moisture, temperature, and humidity. Data collected is transmitted to a cloud-based platform for analysis and automated irrigation scheduling. an innovative approach to modernizing agricultural practices using the Internet of Things (IoT). The paper highlights the design and development of an intelligent system that automates irrigation and monitors crop health, enhancing water efficiency and boosting crop productivity. The system integrates IoT-based sensors to collect real-time data on critical environmental parameters such as soil moisture, temperature, and humidity. This data is processed and analyzed to make informed decisions regarding irrigation schedules, ensuring water is provided only when necessary. The paper further discusses the incorporation of wireless communication and cloud-based platforms for seamless data access and remote monitoring. By utilizing advanced technologies, the system reduces manual effort, minimizes resource wastage, and promotes sustainable farming practices. Additionally, the research includes performance analysis and experimental results that demonstrate the system's effectiveness in optimizing water use and improving crop yield. This IoT-based solution aims to empower farmers with actionable insights, leading to smarter and more efficient agricultural operations.

M. A. Khan Patil is the Author of the paper title "Smart Irrigation Systems in Agriculture: An Overview"published in ResearchGate, 2021 this paper provides a comprehensive review of smart irrigation systems that integrate IoT and machine learning (ML) technologies to optimize irrigation processes. It discusses various case studies and emerging trends in agricultural automation. comprehensive analysis of the role of smart irrigation technologies in modern agriculture. The paper emphasizes the need for sustainable water management practices in agriculture to address the challenges posed by water scarcity and inefficient irrigation methods. It explores various smart irrigation systems that leverage technologies such as the Internet of Things (IoT), sensors, wireless communication, and automated control systems to optimize water usage. The paper discusses different types of sensors, including soil moisture, temperature, and humidity sensors, which provide real-time data for precision irrigation. M. A. Khan highlights the benefits of integrating smart irrigation with data analytics and cloud computing to ensure accurate decision-making and resource management. Furthermore, the paper reviews case studies and examples showcasing the practical implementation and advantages of smart irrigation systems, such as reduced water consumption, improved crop health, and increased agricultural productivity. The study underscores the potential of these systems to revolutionize farming by promoting sustainable practices, enhancing efficiency, and reducing manual labor.

Y. SaikaiPatil is the Author of the paper title"Deep Reinforcement Learning for Irrigation Scheduling Using High-Dimensional Sensor Feedback" published in arXiv, 2023 This paper introduces a deep reinforcement learning (DRL)-based approach for optimizing irrigation schedules. The system uses high-dimensional feedback from soil and environmental sensors to make irrigation decisions dynamically. comprehensive analysis of the role of smart irrigation technologies in modern agriculture. The paper emphasizes the need for sustainable water management practices in agriculture to address the challenges posed by water scarcity and inefficient irrigation methods. It explores various smart irrigation systems that leverage technologies such as the Internet of Things (IoT), sensors, wireless communication, and automated control systems to optimize water usage. The paper discusses different types of sensors, including soil moisture, temperature, and humidity sensors, which provide real-time data for precision irrigation. M. A. Khan highlights the benefits of integrating smart irrigation with data analytics and cloud computing to ensure accurate decision-making and resource management. Furthermore, the paper reviews case studies and examples showcasing the practical implementation and advantages of smart irrigation systems, such as reduced water consumption, improved crop health, and increased agricultural productivity. The study underscores the potential of these systems to revolutionize farming by promoting sustainable practices, enhancing efficiency.

2.1 SUMMARY

The Intelligent Irrigation and Crop Monitoring System integrates IoT, AI, and sensor technologies to improve agricultural productivity by optimizing water usage and monitoring crop health. Various studies have demonstrated significant advancements in this field.

IoT-based irrigation systems, such as the one developed by R. Gutiérrez et al. (2014), utilized soil moisture sensors and GSM technology for remote monitoring, reducing water consumption by 40% without affecting crop yields. Similarly, S. Patel and
D. N. Doshi (2016) introduced a system that used sensors to monitor temperature, humidity, and soil pH, facilitating early detection of crop health issues and improving pest and nutrient management.

A. Sharma et al. explored the scalability of IoT systems using LoRaWAN, achieving cost savings and water efficiency in large-scale farms. Meanwhile, P. Kumar et al. (2020) combined weather forecasting data with soil sensors to develop a weather-aware irrigation system, reducing water wastage by 30%. Energy efficiency was a focus of M. Yadav et al. who designed a solar-powered irrigation system for rural areas, offering a sustainable solution in regions with unreliable electricity

In terms of crop health monitoring, V. Singh and P. Roy (2021) implemented AI-based image processing and drones for early disease detection, improving yields by 15%. T. Chen et al. advanced greenhouse monitoring by automating irrigation and climate control, leading to enhanced crop productivity.

In conclusion, intelligent irrigation and crop monitoring systems have the potential to revolutionize agriculture by conserving resources, enhancing productivity, and supporting sustainable farming practices. Continued innovation and scalability efforts will be critical for their broader implementation.

CHAPTER 3

EXISTING SYSTEM

3.1 INTRODUCTION

The existing systems for irrigation and crop monitoring primarily focus on automating water delivery and collecting environmental data to enhance agricultural efficiency. Traditionally, irrigation methods relied on manual scheduling or fixed systems such as flood irrigation and sprinkler setups, which often result in water wastage and inefficient resource utilization. With advancements in technology, current systems incorporate various tools such as sensors, microcontrollers, and wireless communication to improve water management and monitor crop health.

Many existing systems use soil moisture sensors, temperature sensors, and humidity sensors to collect field data and determine when irrigation is needed. Microcontrollers like Arduino and Raspberry Pi process this data and automate the water flow based on pre-defined thresholds. Additionally, IoT (Internet of Things)- enabled systems have been introduced, allowing real-time data collection and remote monitoring through cloud-based platforms or mobile applications. These systems provide farmers with actionable insights and control over irrigation processes from anywhere.

Despite these improvements, existing systems often have limitations, such as high costs, complex installations, and insufficient adaptability to varying agricultural needs. Some systems may lack advanced decision-making capabilities, relying on static thresholds instead of dynamic environmental factors. Furthermore, the integration of high-dimensional sensor feedback and real-time learning, like Artificial Intelligence (AI) or Deep Learning, is still in its early stages. As a result, there is a growing need for more intelligent, scalable, and cost-effective systems

3.2 TRADITIONAL IRRIGATION SYSTEMS

Traditional irrigation methods, such as flood irrigation and sprinkler systems, have been in use for centuries and form the basis for many agricultural irrigation practices today. However, these systems are often inefficient, particularly in regions with limited water resources.

- Involves flooding the entire field with water, which can result in water evaporation, or uneven distribution.
- This method doesn’t consider soil moisture levels or the crop's water needs, leading to either over-irrigation or under-irrigation.
- Water is sprayed over the crops similar to rainfall, and while it is more efficient than flood irrigation, it can still be wasteful if water is sprayed during windy conditions or if there is uneven distribution across the field.
- Sprinklers often do not account for varying soil moisture content or crop- specific requirements, leading to inefficient water usage.

3.3 SENSOR-BASED IRRIGATION SYSTEMS

With advancements in technology, irrigation systems now incorporate sensors to collect real-time data from the environment and soil, enabling better water management decisions.

driven decision-making tools, existing systems face several notable limitations that hinder their widespread adoption and effectiveness. One primary challenge is the high initial cost of implementation, which includes expenses for purchasing sensors, installing connectivity infrastructure, and integrating advanced software platforms. This financial barrier can be particularly prohibitive for small-scale farmers or organizations with limited budgets.

3.4 IOT-BASED IRRIGATION SYSTEMS

IoT-based irrigation systems are a groundbreaking innovation in the agricultural sector, designed to tackle critical issues such as water scarcity, inefficient resource use, and fluctuating crop productivity. These systems harness the power of the Internet of Things (IoT) by integrating smart devices like soil moisture sensors, temperature gauges, weather monitoring stations, and even drone technology to gather and analyze real-time data from the fields. This data is processed using cloud-based platforms, where advanced algorithms evaluate various parameters, including soil conditions, weather forecasts, crop type, and growth stages, to make data-driven irrigation decisions.

A central feature of IoT-based irrigation is its ability to automate water distribution. Traditional irrigation methods often rely on manual operations or fixed schedules that do not account for real-time field conditions, leading to overwatering or under-irrigation. IoT systems overcome these challenges by enabling precise water delivery exactly when and where it is needed. For example, sensors placed in the soil can detect moisture levels and trigger irrigation only when the soil dries below a predefined threshold. This ensures that crops receive adequate hydration without wasting water, conserving this precious resource in water-stressed regions.

Moreover, IoT-based systems allow farmers to monitor and control irrigation processes remotely via smartphones, tablets, or computer interfaces. This remote access not only saves time and labor but also allows for better decision-making, especially during emergencies or extreme weather conditions. Some advanced systems are integrated with machine learning models that predict future water requirements based on historical data, crop type, and anticipated weather conditions, further enhancing their efficiency.

3.5 ADVANCED DECISION-MAKING AND ARTIFICIAL INTELLIGENCE (AI)

Advanced decision-making, empowered by Artificial Intelligence (AI), is transforming the way organizations and individuals solve complex problems. AI systems utilize vast amounts of data, advanced algorithms, and machine learning techniques to analyze patterns, predict outcomes, and recommend optimal solutions. Unlike traditional methods, AI-driven decision-making is capable of processing information at unprecedented speeds, enabling real-time responses to dynamic situations. For instance, in industries like healthcare, AI assists doctors in diagnosing diseases by analyzing medical records, imaging data, and genetic information. Similarly, in finance, AI models evaluate market trends and predict stock movements with remarkable accuracy.

Beyond data analysis, AI incorporates natural language processing and sentiment analysis to understand human emotions and behavior, further refining decision-making processes. Tools like neural networks and deep learning allow AI systems to identify hidden patterns and correlations that might escape human analysis. Moreover, these systems continuously learn and adapt from new data, improving their accuracy and reliability over time.

The integration of AI in decision-making also minimizes human biases, promotes objectivity, and enhances efficiency. Businesses use AI to optimize supply chains, manage resources, and forecast demand, saving time and reducing costs. In governance, AI-driven tools help policymakers make data-informed decisions to address social and economic challenges.

While AI holds immense potential, it also requires ethical considerations, transparency, and accountability to ensure fair and responsible use. Nonetheless, as AI technology continues to evolve, it is poised to revolutionize decision-making across all domains, unlocking opportunities for innovation and progress in unprecedented ways.

3.6 LIMITATIONS OF EXISTING SYSTEMS

Despite the significant advancements and benefits offered by IoT-based irrigation systems and AI-driven decision-making tools, existing systems face several notable limitations that hinder their widespread adoption and effectiveness. One primary challenge is the high initial cost of implementation, which includes expenses for purchasing sensors, installing connectivity infrastructure, and integrating advanced software platforms. This financial barrier can be particularly prohibitive for small-scale farmers or organizations with limited budgets. Additionally, the technical complexity of these systems requires specialized knowledge for setup, maintenance, and troubleshooting, often necessitating ongoing training and support that may not be readily accessible.

Connectivity issues also pose a significant limitation, especially in rural or remote areas where reliable internet access is inconsistent or unavailable. Without stable connectivity, the real-time data transmission essential for the optimal functioning of IoT and AI systems can be severely disrupted, leading to delays in decision-making and reduced system efficiency. Data security and privacy are further concerns, as the extensive data collection involved in these systems makes them vulnerable to cyber-attacks and unauthorized access, potentially compromising sensitive agricultural information.

Scalability remains another critical issue, as existing systems may struggle to accommodate the diverse and expanding needs of large agricultural operations or varied environmental conditions. Additionally, the dependency on continuous power supply and the durability of sensors in harsh outdoor environments can affect the long-term reliability and sustainability of these technologies. Interoperability between different devices and platforms is often limited, leading to integration challenges that can complicate the seamless operation of comprehensive smart farming solutions.

Moreover, while AI-driven decision-making tools offer enhanced predictive capabilities, they are not infallible and can sometimes produce inaccurate recommendations due to biases in the training data or unforeseen environmental variables. This reliance on data quality and algorithmic accuracy necessitates rigorous validation and continuous refinement to maintain trust and effectiveness. Lastly, the rapid pace of technological advancements can result in compatibility issues and the need for frequent system upgrades, further adding to the operational costs and complexity.

In summary, while IoT-based irrigation systems and AI-driven decision-making tools hold great promise for revolutionizing agriculture and other industries, addressing these limitations is essential to fully realize their potential. Overcoming financial barriers, improving connectivity, ensuring data security, enhancing scalability and reliability, and fostering interoperability are crucial steps toward developing more robust and accessible smart systems that can sustainably meet the diverse needs of users worldwide.

CHAPTER 4

PROPOSED SYSTEM

4.1 THEORETICAL BACKGROUND

Machine learning is an application of artificial intelligence (AI) that gives systems the ability to automatically learn and evolve from experience without being specially programmed by the programmer. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The main aim of machine learning is to allow computers to learn automatically and adjust their actions to improve the accuracy and usefulness of the program, without any human intervention or assistance. Traditional writing of programs for a computer can be defined as automating the procedures to be performed on input data in order to create output artifacts. Almost always, they are linear, procedural and logical. A traditional program is written in a programming language to some specification, and it has properties like:

- Know that or can control the inputs to the program.
- Can specify how the program will achieve its goal.
- Can map out what decisions the program will make and under what conditions makes them.
- Since know the inputs as well as the expected outputs, can be confident that program will achieve its goal
- Such problems resist a procedural and logical solution. They have properties
- The scope of all possible inputs is not known beforehand.

Machine learning techniques can be broadly categorized into the following types: Supervised learning takes a set of feature/label pairs, called the training set. From this training set the system creates a generalized model of the relationship between the set of descriptive features and the target features in the form of a program that contains a set of rules. The objective is to use the output program produced to predict the label for a previously unseen, unlabelled input set of features, i.e. to predict the outcome for some new data. Data with known labels, which have not been included in the training set, are classified by the generated model and the results are compared to the known labels. This dataset is called the test set. The accuracy of the predictive model can then be calculated as the proportion of the correct predictions the model labeled out of the total number of instances in the test set.

Unsupervised learning takes a dataset of descriptive features without labels as a training set. In unsupervised learning, the algorithms are left to themselves to discover interesting structures in the data. The goal now is to create a model that finds some hidden structure in the dataset, such as natural clusters or associations. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data.

IoT forms the backbone of intelligent irrigation systems by enabling communication between sensors, controllers, and central systems. Sensors are deployed in the field to measure parameters like soil moisture, temperature, humidity, and light intensity. This data is transmitted wirelessly to a central hub or cloud platform for analysis. The theoretical basis lies in the use of communication protocols like ZigBee, LoRa, and Wi-Fi to establish a seamless network for data exchange.

Artificial Intelligence enhances the system's ability to analyze collected data and make informed decisions. Machine learning models, trained on historical and real-time data, predict crop water requirements, detect anomalies, and schedule irrigation cycles.

predictive models estimate future water needs based on weather forecasts and crop growth stages.

4.2 FUNCTIONAL REQUIREMENTS:

- System must be fast and efficient
- User friendly GUI
- Performance
- System Validation input
- Proper output

System Feature 1(Functional Requirement):

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality. Following are the functional requirements on the system:

- All the data must be in the same format as a structured data.
- The data collected will be vectorized and sent across to the classifier.

External Interface Requirements:

The external interface requirements for the proposed intelligent irrigation and crop monitoring system cover hardware integration (sensors, actuators), software interactions (cloud, AI/ML platforms, mobile/web apps), user interactions, external system integrations (weather APIs, agricultural data), and security measures. These interfaces ensure that the system can effectively gather, process, and transmit data, providing farmers with actionable insights while ensuring reliability, security, and scalability for diverse agricultural needs.

Front End Software: Flask Framework integrated with HTML, CSS, JS

Flask is a lightweight Python web framework that seamlessly integrates with front-end technologies like HTML, CSS, and JavaScript to create dynamic and interactive web applications. At its core, Flask uses the Jinja2 templating engine, which allows developers to embed Python code into HTML templates for dynamic content rendering. HTML forms the structure of the web pages, while CSS is used for styling, enhancing the visual appeal, and improving user experience.

Flask serves static files, such as CSS and JavaScript, from a dedicated /static directory, enabling easy integration of front-end assets. JavaScript adds interactivity, allowing for responsive features like form validation and dynamic updates without refreshing the page. Flask can handle AJAX requests, enabling real-time communication between the server and client, making it possible to fetch or send data dynamically. This synergy between Flask and front-end technologies empowers developers to build modern web applications that are both functional and visually engaging, meeting the demands of users efficiently.

Back End Software: Machine Learning (Python)

Machine Learning (ML) with Python serves as a robust backbone for backend software, enabling intelligent systems to analyze data, make predictions, and automate decision-making processes. Python’s simplicity and extensive library support, such as TensorFlow, Scikit-learn, PyTorch, and Pandas, make it a popular choice for implementing machine learning algorithms. In the backend, ML models process large datasets to extract patterns, classify information, and predict outcomes. For instance, a recommendation system can analyze user behavior to suggest products, while predictive analytics can forecast trends in data-intensive.

Software Interfaces:

- OS: Ubuntu, Windows, Mac
- Tools: VScode or Python IDE and Jupyter Notebook.
- Programming Language: Python flask, HTML, CSS, JS, BOOTSTRAP.
- Dataset: A Dataset which is openly available in kaggle

Communication Interfaces:

The communication protocol started from http from UI interface in which person can see some details of the soil nutrition section by filling up details user can see which type of crop.

Nonfunctional Requirements:

Nonfunctional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the requirements arise through the user needs, because of budget constraints, organizational policies and the need for interoperability with other software and hardware systems.

Project Scope:

- Improve farm management efficiency by adjusting field/crop treatments
- Getting a better result for which type of crop will be growing on your agriculture
- Getting more productivity from less efforts by using our application
- Improve farm management efficiency by adjusting field/crop treatments
- Optimize efforts and resources, reduce consumption and waste, and boost land
- Which type of fertilizers should be used if any crop having any disease we minimize using our app.

Performance Requirements:

An intelligent irrigation and crop monitoring system must meet several performance requirements to ensure efficient, reliable, and sustainable agricultural operations. The system should process sensor data, such as soil moisture, temperature, and humidity, in real time to enable prompt decision-making and irrigation control. High accuracy is essential for precise monitoring of environmental and soil conditions, as well as detecting anomalies in crop health. Scalability is another critical factor, allowing the system to handle large fields or multiple sites without compromising performance. Low latency is crucial to ensure minimal delays in communication between sensors, actuators, and central processing units, enabling timely responses.

Energy efficiency is vital, especially for sensors and IoT devices deployed in remote areas, requiring optimized hardware and algorithms to minimize power consumption. Reliable connectivity using protocols like ZigBee, LoRa, or Wi-Fi ensures uninterrupted data transmission even in challenging environments. The system must also support efficient data storage and quick retrieval for historical analysis and predictive modeling. Fault tolerance and reliability are necessary to handle hardware or software failures gracefully, ensuring continuous operation with minimal downtime.

Adaptability to changing weather conditions, such as rainfall or temperature fluctuations, allows the system to dynamically adjust irrigation schedules. A user-friendly interface is essential for providing clear insights and controls, enabling operators or farmers to interact with the system intuitively. Strong security measures, including encryption and authentication, protect sensitive agricultural data from unauthorized access.

Methodology:

The methodology for developing the Intelligent Irrigation and Crop Monitoring System emphasizes an iterative approach involving system design, sensor integration, data processing, cloud-based storage, machine learning for predictive analysis, and automated irrigation control. The system aims to optimize water usage, improve crop health, and ensure sustainability by leveraging modern technologies. Through continuous monitoring and adaptive learning, the system provides farmers with the tools necessary to manage irrigation efficiently and respond to changing environmental conditions in real- time.

The proposed system aims to establish a comprehensive end-to-end farming solution that integrates an efficient automatic irrigation subsystem and a crop disease classifier. The system leverages IoT for data collection and control and employs a Convolutional Neural Network (CNN) to classify crop diseases.

In the Automatic Irrigation System, soil moisture sensors monitor the moisture levels in the soil. Based on these readings, the system automatically irrigates the crop according to its specific moisture requirements, ensuring optimal water use. Simultaneously, the system incorporates an automated mechanism to capture images of the crop at regular intervals. These images are synced to the Deep Learning Model, where a pre-trained CNN processes the input. The model analyzes the images to classify the leaf health into one of three categories: Healthy, Late Blight, or Early Blight.

The Intelligent Irrigation and Crop Monitoring System is designed to optimize agricultural practices by integrating IoT, sensor networks, and deep learning techniques. The system comprises two primary subsystems: an Automatic Irrigation System and a Crop Disease Classification Model, both interconnected to function as an end-to-end smart farming solution. The following methodology outlines the workflow, technologies, and operational principles of the system.

The Automatic Irrigation System uses IoT-based sensors to monitor soil moisture levels in real time. These sensors are strategically placed in the field to gather accurate data on soil conditions. The collected data is sent to a central microcontroller, such as Arduino or Raspberry Pi, which processes the inputs using predefined thresholds and decision-making algorithms.

Simultaneously, the system includes a mechanism for capturing periodic images of the crops. These images are transferred to a cloud-based platform, where they are analyzed by the Deep Learning Model for disease detection. A Convolutional Neural Network (CNN) is employed for image classification, trained on a dataset containing images of healthy plants and plants affected by diseases such as Late Blight and Early Blight. The model processes the images and classifies the leaves into three categories: Healthy, Late Blight, or Early Blight. This classification is critical for early detection of diseases, enabling timely interventions to prevent further spread and potential crop loss.

The integration of the two subsystems is achieved through automated synchronization. The irrigation system is equipped with a camera that captures images during or after irrigation cycles. These images are sent as test data to the CNN-based model for analysis. The results from the model are relayed to a user-friendly dashboard remotely. Notifications and alerts are generated for conditions requiring immediate attention, such as severe moisture deficits or disease detection.

Cloud computing supports the storage and processing of large volumes of data collected from sensors. It provides computational resources for running complex AI algorithms and facilitates remote access to system insights through mobile or web-based interfaces.

To ensure energy efficiency and sustainability, the system is often powered by renewable energy sources like solar panels, making it suitable for deployment in rural areas with limited access to electricity. The entire system operates in a closed loop, where feedback from the disease classification and soil moisture monitoring continuously refines irrigation schedules and guides the farmer on necessary crop.

Feasibility Study:

Analysis is the process of finding the best solution to the problem. System analysis is the process by which we learn about the existing problems, define objects and requirements and evaluates the solutions.

Economical Feasibility:

This study is carried out to check the economic impact that the system will have on the organization. Since the project is Machine learning based, the cost spent in exectuating this project would not demand cost for softwares and related products, as most of the products are open source and free to use. Hence the project would consumed minimal cost and is economically feasible.

Technical Feasibility:

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Since machine learning algorithms is based on pure math there is very less requirement for any professional software. And also most of the tools are open source. The best part is that we can run this software in any system without any software requirements which makes them highly portable. Also most of the documentation and tutorials make easy to learn the technology.

Social Feasibility:

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The main purpose of this project which is based on crop prediction is to prevent the farmer from incurring losses and improve productivity. This also ensures that there is no scarcity of food as lack of production may lead to severe consequences. Thus, this is a noble cause for the sake of the society, a small step taken to achieve a secure future.

4.3 SYSTEM DESIGN

The system design for the Intelligent Irrigation and Crop Monitoring System involves a comprehensive structure that integrates hardware, software, and communication technologies to automate irrigation processes and provide real-time monitoring and management of crops. The design is modular and scalable, allowing easy updates and extensions based on specific requirements**.**

Analysis Models: SDLC Model to be applied:

The waterfall model is a sequential software development process, in which progress is seen as owing steadily downwards (like a waterfall) through the phases of Requirement initiation, Analysis, Design, Implementation, Testing and maintenance.

Requirement Analysis:

This phase is concerned about collection of requirement of the system. This process involves generating document and requirement review.

System Design:

Keeping the requirements in mind the system specifications are translated in to a software representation. In this phase the designer emphasizes on:- algorithm, data structure, software architecture etc.

Coding:

In this phase programmer starts his coding in order to give a full sketch of product. In other words system specifications are only converted in to machine.

Implementation:

The implementation phase involves the actual coding or programming of the software.

Waterfall Model :

The Waterfall Model is a linear and sequential software development process that is structured in distinct phases. Each phase must be completed before the next phase can begin. The Waterfall Model is best suited for projects were the

Are well understood from the beginning and are unlikely to change during the development process

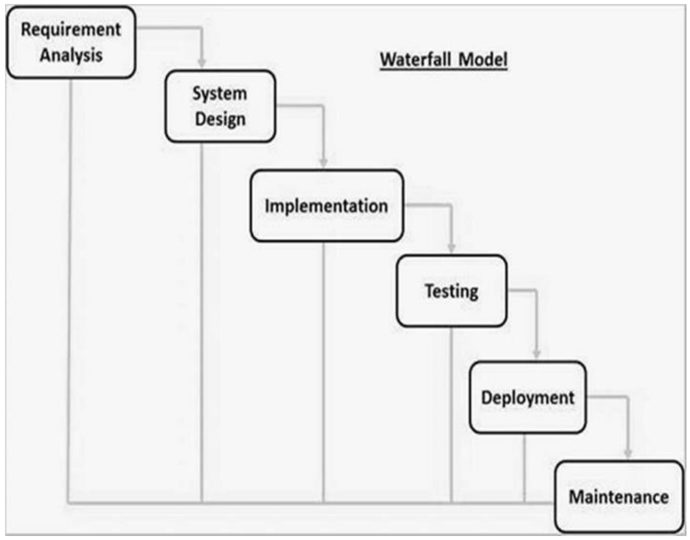

Figure 4.1 Waterfall Model

System Architecture:

The system architecture of the Intelligent Irrigation and Crop Monitoring System defines the structure of various components and how they interact to achieve the desired functionality. The architecture ensures that data flows seamlessly between hardware components, software platforms, and the user interface, facilitating real-time monitoring and automated irrigation based on sensor inputs.

User Interface Layer

- Mobile Application: Provides a user-friendly interface for farmers to:
- View real-time data such as soil moisture levels, temperature, and humidity.
- Receive alerts regarding irrigation needs or system faults.
- Control irrigation manually (on/off or adjust settings).
- Monitor historical data and trends.

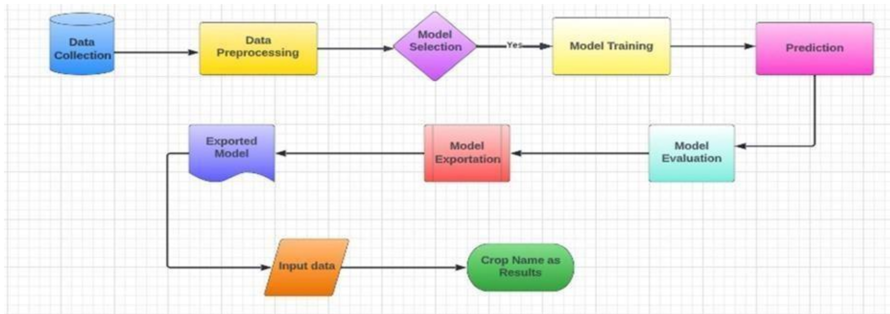

Figure 4.2 System Architecture

Sequence Diagram:

A Sequence Diagram represents the interaction between various system components in a sequential manner. It shows how objects interact with each other in a particular sequence of events to complete a specific process or function.

In the context of the Intelligent Irrigation and Crop Monitoring System, the sequence diagram describes how different components (sensor, microcontroller, cloud, user interface) interact during a typical irrigation cycle.

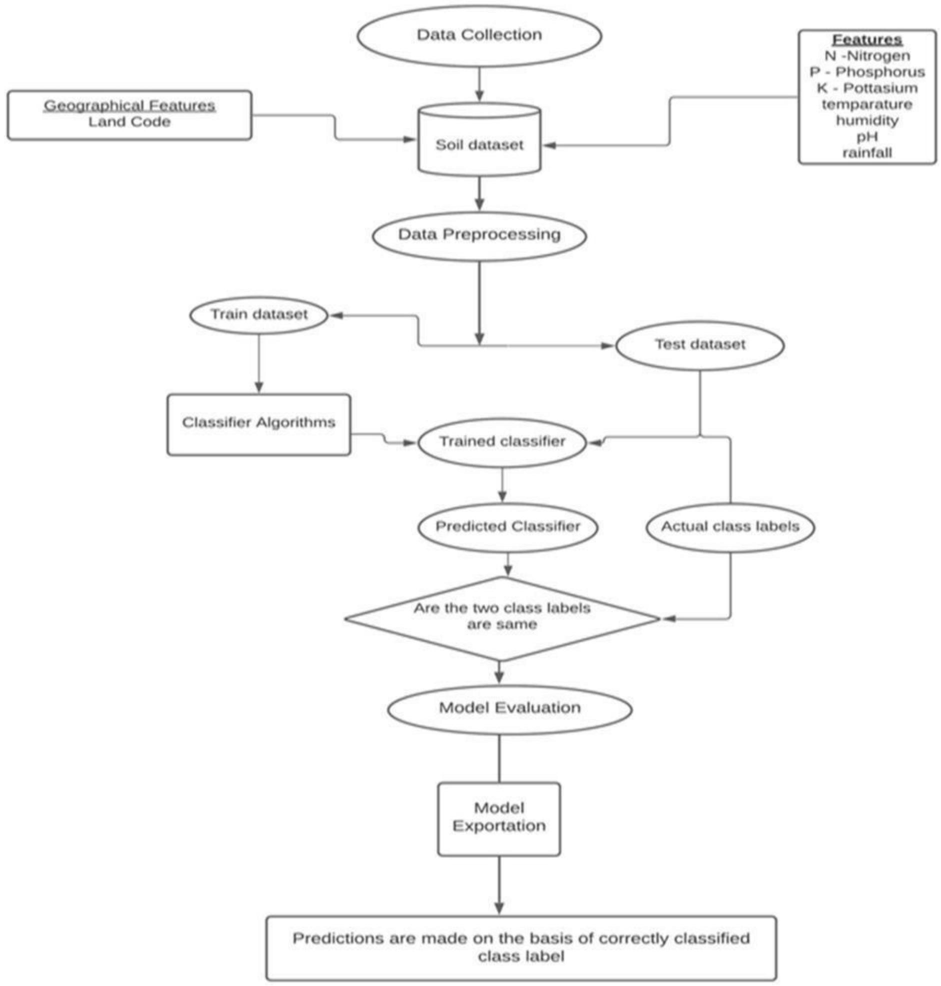

Figure 4.3 Sequence Diagram

DFD Level-0:

The Data Flow Diagram (DFD) Level 0 represents the high-level functional processes of the Intelligent Irrigation and Crop Monitoring System. This diagram abstracts the interactions between the system's components and external entities, offering a clear representation of the main data flows and processes within the system.

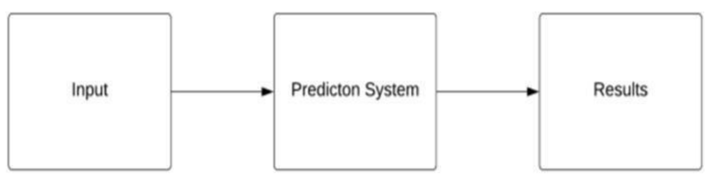

Figure 4.4 DFD Level-0

DFD Level-1:

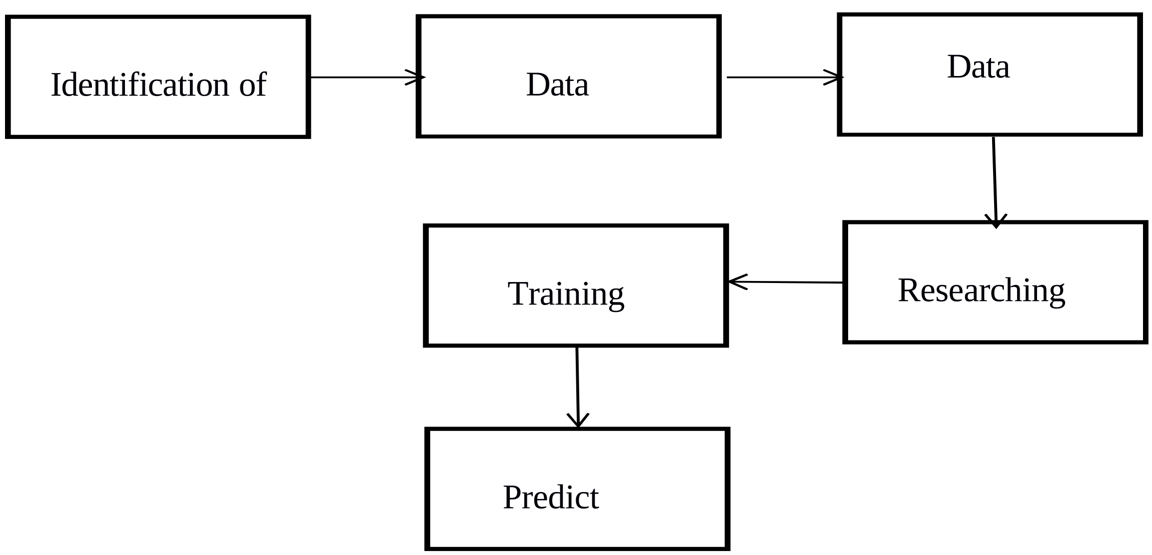

Figure 4.5 DFD Level 1

Level 1 DFDs are still a general overview, but they go into more detail than a context diagram. In level 1 DFD, the single process node from the context diagram is broken down into sub-processes. As these processes are added, the diagram will need additional data flows and data stores to link them together. In the hotel reservation example, this can include adding the room selection and inquiry processes to the reservation system, as well as data stores.

UML Diagrams:

A UML diagram for an intelligent irrigation and crop monitoring system visually represents the system's design and functionality. It includes key elements such as sensors, controllers, actuators, and user interfaces, along with their interactions. For instance, in a Use Case Diagram, the main actors like farmers, sensors, and external weather APIs interact with the system through use cases such as monitoring soil conditions, analyzing weather data, controlling irrigation equipment, and receiving notifications.

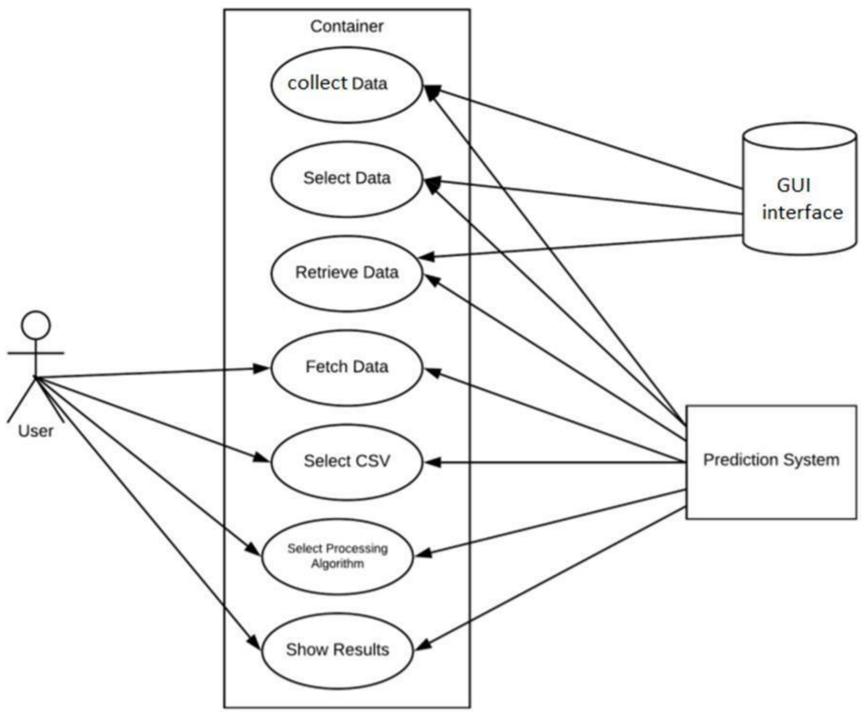

Figure 4.6 UML diagram

Table 4.1 Yield Dataset

A1 | state

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	state	Rice	Jowar	Bajra	Maize	Ragi	Wheat	Barley	Gram	Tur	Groundnu	Mustard	Soyabean	Sunflowe	Cotton	Jute	Mesta	Sugarcan
2	Andhra Pr	2921.186	1054.115	906.734	3328.871	1202.798	792.5525	0	1139.872	441.4852	852.6383	356.25	1450.464	768.493	337.1242	0	1507.329	76472.3
3	Arunachal	1190.479	0	0	1365.529	0	1641.025	0	0	1000	0	963.015	1258.428	0	0	0	0	19073.6
4	Assam	1449	0	0	710	0	1150	0	519	707	0	508	0	0	106	1736	909	3794
5	Bihar	1300.151	929.1806	933.166	2281.541	794.861	1876.421	1124.002	938.5558	1183.194	652.7778	853.5547	0	1365.018	0	1474.782	1344.15	43134.9
6	Chattisgar	1177	843	0	1567	263	996	888	728	471	1122	369	857	459	136	0	361	255
7	Goa	2652	0	0	0	1000	0	0	0	0	1813	0	0	0	0	0	0	5216
8	Gujarat	1610	901	1152	1300	868	2451	0	739	796	1161	1341	716	0	373	0	0	7405
9	Haryana	2894	296	1313	2228	0	3979	2735	725	988	809	1304	0	1598	452	0	0	5998
10	Himachal	1447	0	0	2251	1104	1482	1207	901	0	0	495	1342	0	0	0	0	1801
11	Jammu &	1960	589	571	1535	0	1543	631	0	0	0	635	0	0	0	0	0	0
12	Jharkhand	1413	988	1253	1465	632	1682	922	886	860	698	558	0	0	0	0	1051	3460
13	Karnataka	2561.393	845.2645	626.1794	2654.594	1492.443	736.6376	0	506.7264	489.3979	696.5809	278.5281	683.0272	456.3873	219.2412	0	265.7269	83235.9
14	Kerala	2197	490	0	0	1070	0	0	0	0	763	0	0	0	250	0	0	9136
15	Madhya P	862	985	1244	1525	351	1630	1228	855	754	992	925	928	453	164	0	382	3893
16	Maharash	1594	812	695	1835	992	1320	637	614	683	1066	317	1175	534	189	0	273	7520
17	Manipur	2315.246	0	0	2495.178	0	0	0	0	0	0	461.1111	0	0	0	0	0	32206.88
18	Meghalay	1692	0	0	1452	0	1699	0	0	769	0	648	945	0	172	1430	835	0
19	Mizoram	1501	0	0	1814	0	0	0	0	0	0	742	1113	0	368	0	0	407
20	Nagaland	1556	1246	1269	1609	0	1716	1756	1027	992	1308	842	1282	1197	570	587	0	4475
21	Orissa	1366.031	608.9923	559.4005	1412.742	641.3716	1445.9	0	627.3451	704.1701	1111.208	205.4544	769.2308	802.4356	327.1654	1797.091	796.46	60066.4
22	Punjab	3686	0	984	2702	0	4259	3309	892	876	868	1105	0	1602	563	0	0	6027

Table 4.2 Temperature Rainfall and Nutrients dataset

A1 | Bajra

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Bajra	3	18	30	3	8	350	750	L	L	M		
2	Banana	4	15	35	6.5	8.5	450	750	M	VL	VL		
3	Barley	4	12	32	3	8	800	1100	VL	VL	M		
4	Bean	2	14	32	5.5	6.5	300	500	L	VL	M		
5	Black pep	6	23	33	5.5	6.5	1200	2500	H	VL	M		
6	Blackgram	2	23	35	5	7	500	700	L	H	VL		
7	Bottle Gou	2	24	27	6.5	7.5	400	650	VL	VL	VL		
8	Brinjal	3	15	32	5.5	6.5	600	1000	VL	L	M		
9	Cabbage	4	12	30	5.5	6.5	300	600	M	VL	H		
10	Cardamon	8	18	35	4.5	7	1200	4000	H	M	M		
11	Carrot	4	7	23	5.5	7	750	1000	M	H	M		
12	Castor see	6	20	30	5	8.5	500	800	VL	H	VL		
13	Cauliflow	4	12	30	6	7	100	300	M	M	M		
14	Chillies	3	18	40	5.5	7	625	1500	VL	VL	L		
15	Coriander	3	15	30	6	10	750	1000	L	L	M		
16	Cotton	4	15	35	6	8	500	1100	M	VL	VL		
17	Cowpea	5	22	35	5	7	700	1100	VL	VL	VL		
18	Drum Stic	4	20	30	6	7	750	2000	M	L	H		
19	Garlic	4	10	30	6	7	500	800	VL	M	H		
20	Ginger	8	15	35	5	7	1200	1800	VL	M	VL		
21	Gram	4	20	30	5	7	600	900	VL	VL	H		
22	Grapes	4	15	35	6.5	8.5	650	850	VL	H	L		
23	Groundnu	3	20	35	5	7	500	750	VL	VL	VL		

Table 4.3 Soil Nutrients distribution as per

crop(Nitrogen,Phosphorous,Potassium).

A	B	C	D	E	F
	Crop	N	P	K	pH
0	rice	80	40	40	5.5
3	maize	80	40	20	5.5
5	chickpea	40	60	80	5.5
12	kidneybe	20	60	20	5.5
13	pigeonpe	20	60	20	5.5
14	mothbear	20	40	20	5.5
15	mungbear	20	40	20	5.5
18	blackgram	40	60	20	5
24	lentil	20	60	20	5.5
60	pomegran	20	10	40	5.5
61	banana	100	75	50	6.5
62	mango	20	20	30	5
63	grapes	20	125	200	4
66	watermel	100	10	50	5.5
67	muskmelc	100	10	50	5.5
69	apple	20	125	200	6.5
74	orange	20	10	10	4
75	papaya	50	50	50	6
88	coconut	20	10	30	5
93	cotton	120	40	20	5.5
94	jute	80	40	40	5.5
95	coffee	100	20	30	5.5

Data Preprocessing:

Data preprocessing is an important step as it helps in cleaning the data and making it suitable for use in machine learning algorithms. Most of the focus in preprocessing is to remove any outliers or erroneous data, as well as handling any missing values.

Missing data can be dealt with in two ways. The first method is to simply remove the entire row which contains the missing or error value. While this an easy to execute method, it is better to use only on large datasets. Using this method on small datasets can reduce the dataset size too much, especially if there are a lot of missing values.

This can severely affect the accuracy of the result. Since ours is a relatively small dataset, we will not be using this method. The dataset that we used had values that were in string format so we had to transform and encode the into integer valued so as to pass as an input to the neural network. First we converted the data into pandas categorical data and then generated codes for crops and states respectively we than appended these and created separated datasets.

crucial step in an intelligent irrigation and crop monitoring system, ensuring raw data from various sources is clean, consistent, and ready for analysis. Data is first gathered from sources such as soil moisture sensors, temperature sensors, humidity sensors, and external weather APIs. Since this raw data is often noisy and incomplete, cleaning techniques are applied to handle missing values and remove outliers, ensuring data reliability. The data is then transformed through normalization or standardization to maintain consistency and scaled for effective analysis. Features relevant to irrigation decisions, such as soil moisture trends or rainfall predictions, are extracted, while data from multiple sources is integrated into a unified format for seamless processing.

```
        decision rules and the fitter the model.

        features = df[['N', 'P','K','temperature', 'humidity', 'ph',
'rainfall']] target = df['label']
        #features = df[['temperature', 'humidity', 'ph',
'rainfall']] labels = df['label']
        # Initialzing empty lists to append all model's name and
corresponding name acc = []
        model = []

        # Splitting into train and test data

        from sklearn.model_selection import train_test_split

        Xtrain, Xtest, Ytrain, Ytest = train_test_split(features,target,test_size =
0.2,random_state =2) from sklearn.tree import DecisionTreeClassifier

DecisionTree.fit(Xtrain,Ytrain)

        predicted_values = DecisionTree.predict(Xtest)
        x = metrics.accuracy_score(Ytest,
predicted_values) acc.append(x)
        model.append('Decision Tree')
print("DecisionTrees's Accuracy is: ", x*100)
```

The business login transaction processor functions should have unit tests, ideally with 100 percent code coverage. This will ensure that you do not have typos or logic errors in the business logic. The various modules can be individually run from a command line and tested for correctness. The tester can pass various values, to check the answer returned and verify it with the values given to him/her. The other work around is to write a script, and run all the tests using it and write the output to a log _le and using that to verify the results.

Dimensionality reduction or sampling is performed if datasets are large, enabling efficient computation. Proper preprocessing not only enhances the accuracy of the system but also ensures robust and reliable decision-making for optimal irrigation and crop monitoring.

```
        from   sklearn.svm   import
SVC SVM = SVC(gamma='auto')
SVM.fit(Xtrain,Ytrain)

        predicted_values = SVM.predict(Xtest)
        x        =        metrics.accuracy_score(Ytest,
predicted_values) acc.append(x)
        model.append('SVM')
print("SVM's Accuracy is: ", x)
        print(classification_report(Ytest,predicted_val
ues))                    score                    =
cross_val_score(SVM,features,target,cv=5)
```

Logistic Regression:

Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Some of the examples of classification problems are Email spam or not spam, Online transactions Fraud or not Fraud, Tumor Malignant or Benign. Logistic regression transforms its output using the logistic sigmoid function to return a probability value.

The types of logistic regression

Binary (eg. Tumor Malignant or Benign)

Multi-linear functions failsClass (eg. Cats, dogs or Sheep's)

Logistic Regression is a Machine Learning algorithm which is used for the classification problems, it is a predictive analysis algorithm and based on the concept of probability.

Average the values for each observation, produced by each tree, if you’re working on a Regression task.

Do a majority vote across all trees, for each observation, if you’re working on a Regression

CHAPTER 5

TESTING

5.1 TESTING METHODOLOGIES:

Testing methodologies in an intelligent irrigation and crop monitoring system are essential to ensure the system's accuracy, reliability, and performance. These methodologies involve a structured approach to evaluate each component, from sensors and communication modules to decision-making algorithms and user interfaces.

One key method is unit testing, where individual components like soil moisture sensors, temperature sensors, or data processing modules are tested independently to verify their functionality. Integration testing follows, focusing on how these components interact, such as the communication between sensors, the controller, and the actuator. This step ensures that data flows seamlessly through the system and triggers appropriate responses.

System testing evaluates the system as a whole, ensuring that it meets functional and performance requirements, such as real-time data processing, accurate irrigation decisions, and reliable notifications. Stress testing is conducted to assess how the system performs under extreme conditions, such as high sensor input rates or network connectivity issues.

Additionally, validation testing ensures that the system operates correctly in real-world scenarios, such as varying weather conditions and soil types. User acceptance testing (UAT) involves farmers or operators to verify the ease of use and practicality of the user interface and notifications. Lastly, security testing checks for vulnerabilities to protect sensitive data and prevent unauthorized access.

5.2 UNIT TESTING:

Unit tests focus on ensuring that the correct changes to the world state take place when a transaction is processed. The business login transaction processor functions should have unit tests, ideally with 100 percent code coverage. This will ensure that you do not have typos or logic errors in the business logic. The various modules can be individually run from a command line and tested for correctness. The tester can pass various values, to check the answer returned and verify it with the values given to him/her. The other work around is to write a script, and run all the tests using it and write the output to a log _le and using that to verify the results. We tested each of the algorithms individually and made changes in preprocessing accordingly to increase the accuracy.

5.3 SYSTEM TESTING:

System Testing is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the systems compliance with the specified requirements. System Testing is the testing of a complete and fully integrated software product and White Box Testing. System test falls under the black box testing category of software testing. Different Types of System Testing:

- **Usability Testing** - Usability Testing mainly focuses on the users ease to use application, exibility in handling controls and ability of the system to meet its objectives.
- **Load Testing** - Load Testing is necessary to know that a software
- **Migration Testing** - Migration testing is done to ensure that the software can from older system infrastructures to current system infrastructures without any issues.

5.4 QUALITY ASSURANCE:

Quality Assurance is popularly known as QA Testing, is defined as an activity to ensure that an organization is providing the best possible product or service to customers. QA focuses on improving the processes to deliver Quality Products to the customer. An organization has to ensure, that processes are efficient and effective as per the quality standards defined for software products.

5.5 FUNCTIONAL TEST:

Functional Testing is also known as functional completeness testing, Functional Testing involves trying to think of any possible missing functions. As chat-bot evolves into new application areas, functional testing of essential chatbot components. Functional testing evaluates use-case scenarios and related business processes, such as the behavior of smart contracts.

Critical process in the development of an intelligent irrigation and crop monitoring system, ensuring that each feature of the system operates according to the specified requirements. This type of testing focuses on verifying the functionality of individual components, such as sensors, data processing algorithms, decision-making modules, and user interfaces. For example, the system's ability to accurately read soil moisture levels, process data in real-time, and activate irrigation equipment when thresholds are met are tested to ensure proper functionality.

Functional testing also evaluates user interactions, such as the farmer's ability to monitor field conditions, receive notifications, and control irrigation manually through the interface. Each function is tested under various conditions to confirm that it performs as expected, even in edge cases or unusual scenarios. By systematically validating that all components and features work together seamlessly, functional testing ensures the system delivers reliable and consistent performance, contributing to efficient and effective irrigation management.

5.6 RESULTS

Algorithms and Their Accuracy:

Decision Trees, Logistic regression, Support Vector Machine and Random Forest. All the algorithms are based on supervised learning. Our overall system is divided
into two modules:

- Crop recommender
- Fertilizer Recommender/Suggestion

Algorithm	Accuracy (%)
Decision Tree	90%
SVM	97%
Logistic Regression	95%
Random Forest	99%

Accuracy Comparison of ML Models:

```
plt.figure(figsize=[10,5],dp
i = 100) plt.title('Accuracy
Comparison') plt.xlabel('Accuracy')
plt.ylabel('Algorithm')
sns.barplot(x = acc,y = model,palette='dark')
```

Output screenshot

Figure 5.1 Welcome page

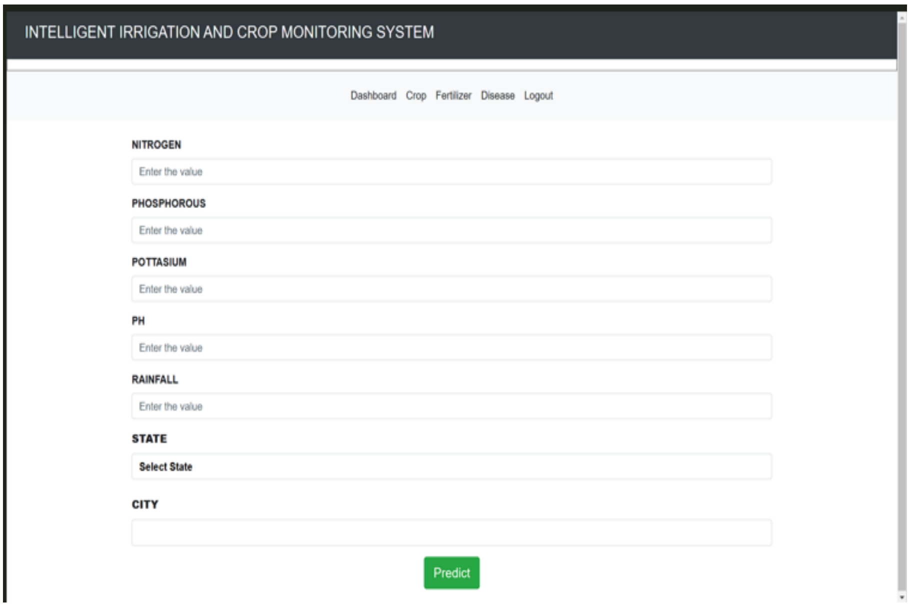

Figure 5.2 Crop recommendation

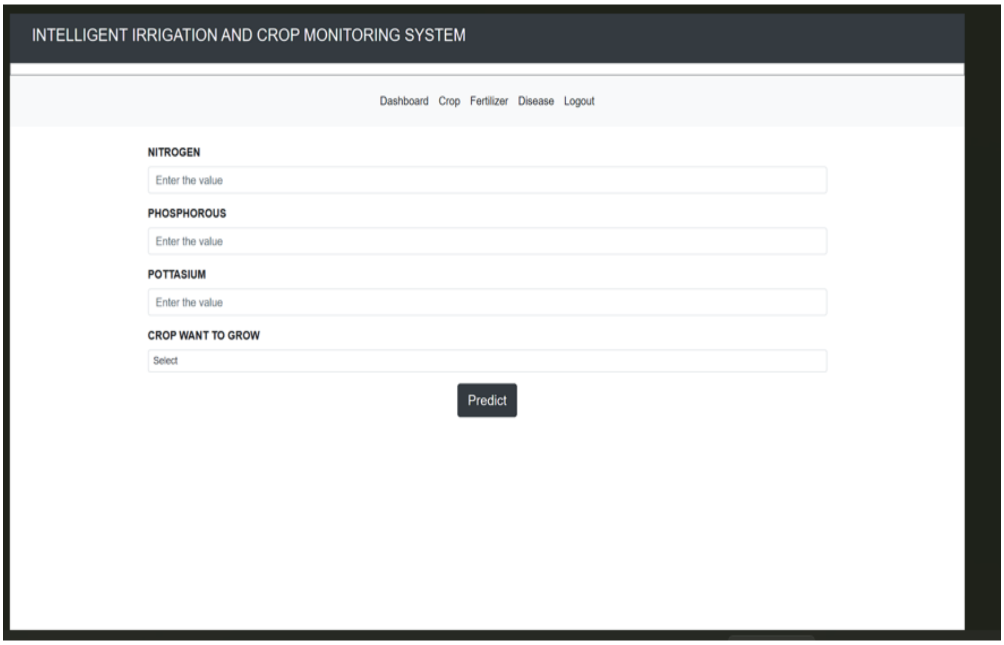

Figure 5.3 Fertilizer recommendation

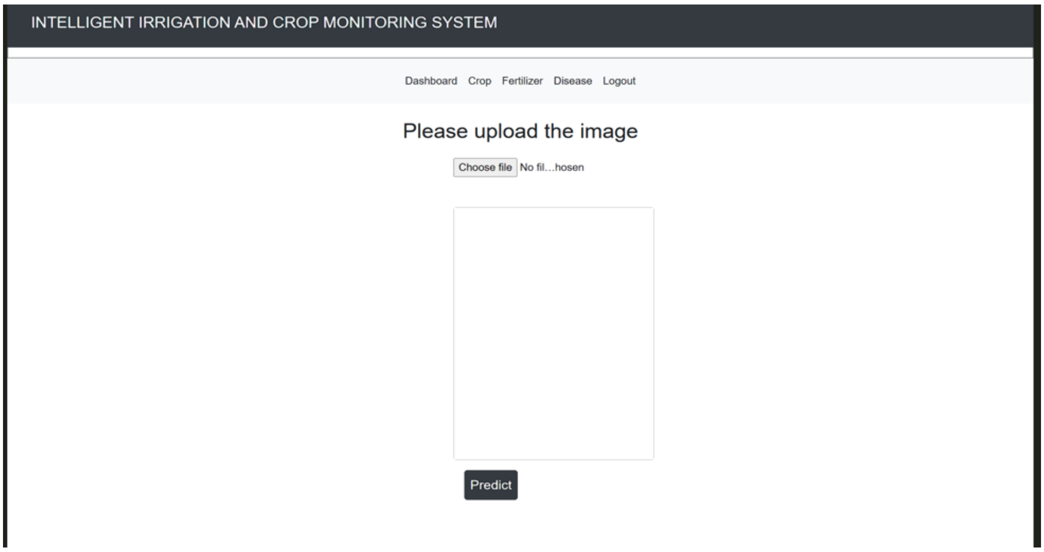

Figure 5.4 Crop disease prediction

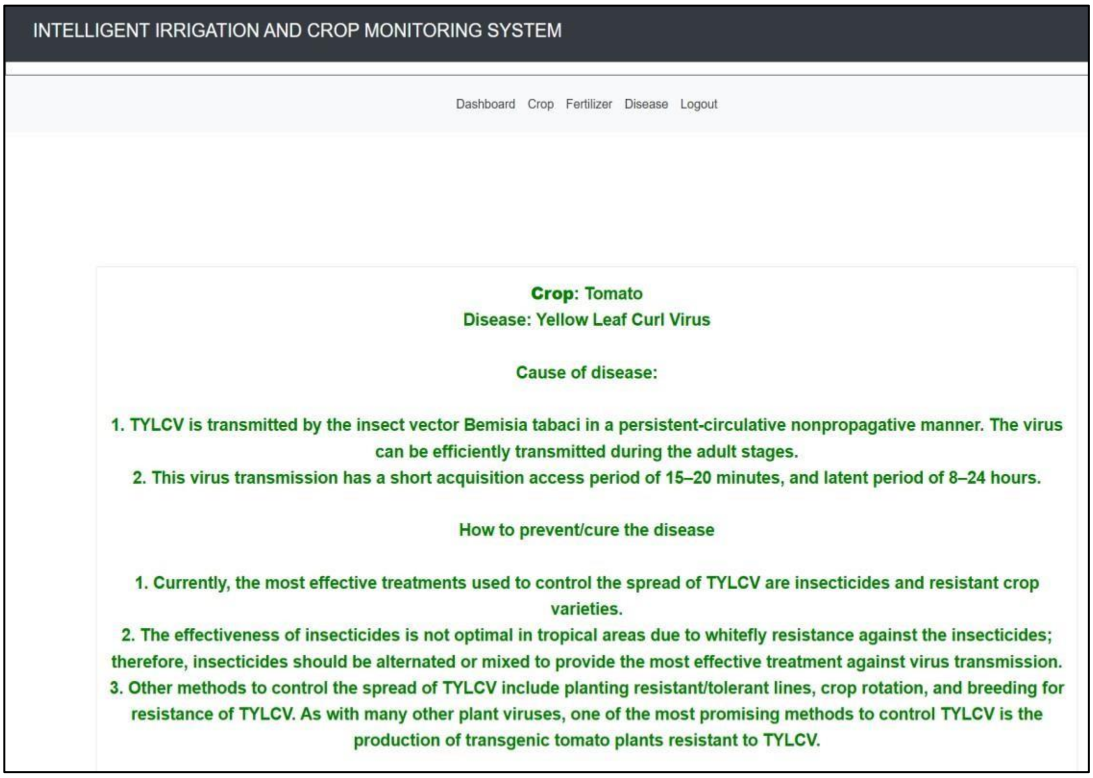

Figure 5.5 Disease suggestion

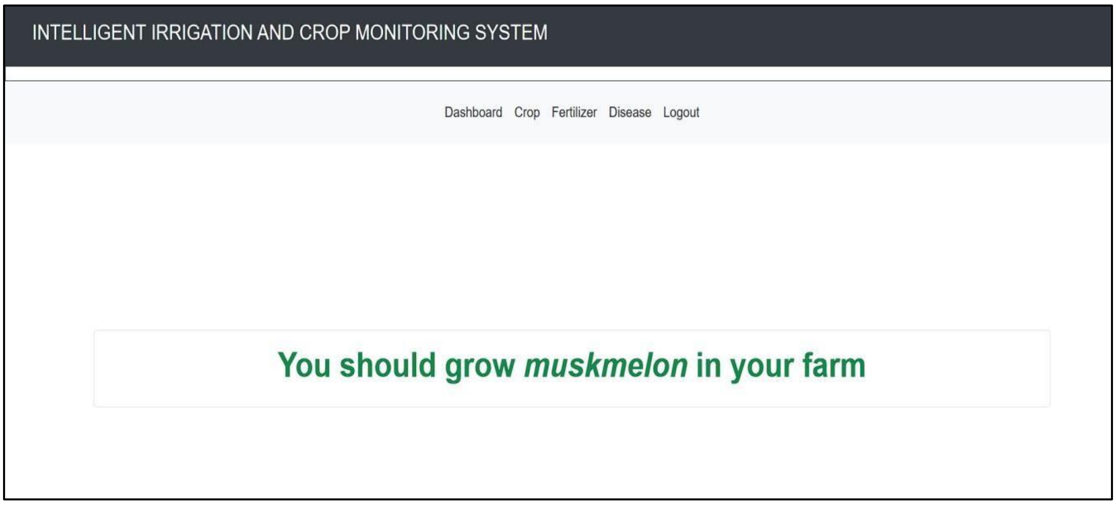

Figure 5.6 Crop recommendation(output)

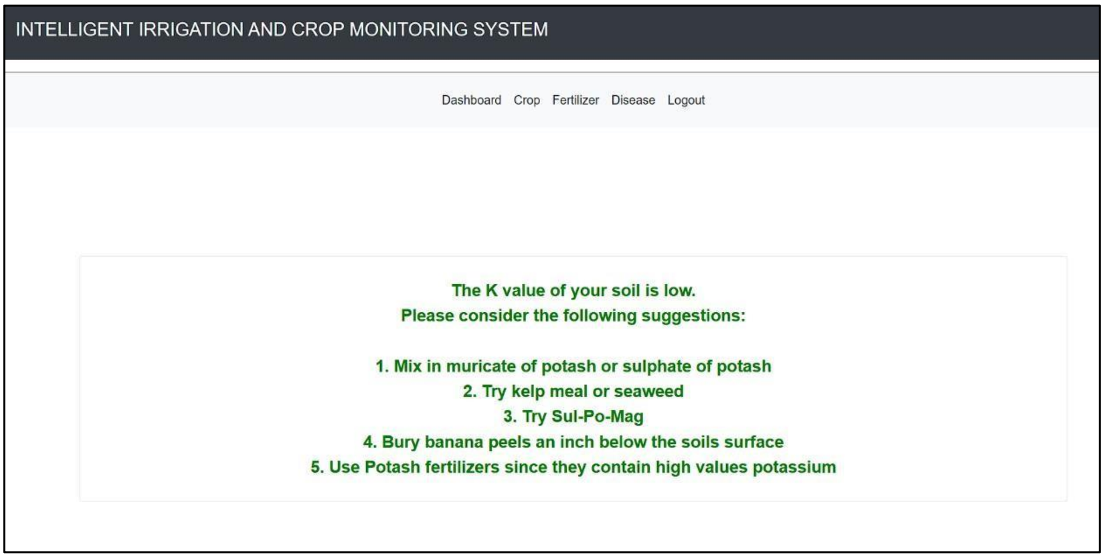

Figure 5.7 Disease description

Advantages:

Water Conservation:

Delivers water only when and where needed, minimizing waste and reducing water consumption.

Increased Yield:

Ensures optimal moisture levels in soil, enhancing plant health and boosting crop production.

Cost-Effectiveness:

Reduces expenses on water and energy by optimizing irrigation schedules.

Remote Monitoring and Control:

Allows farmers to monitor and adjust irrigation settings remotely via smartphones or computers.

Adaptability to Climate Change:

Uses real-time weather data to adjust irrigation patterns, making farming more resilient to unpredictable weather.

Applications:

Precision Farming:

Enables precise control of water distribution based on soil moisture, crop type, and weather data.

Greenhouse Automation:

Maintains optimal conditions by regulating irrigation in controlled environments.

Horticulture and Landscaping:

Ensures efficient water usage for ornamental plants, gardens, and

lawns. Urban Farming:

Supports water-efficient irrigation in vertical farms and urban agricultural

setups. Arid and Semi-Arid Regions:

Minimizes water wastage in regions with scarce water resources by providing targeted irrigation.

Smart Irrigation for Orchards:

Monitors moisture levels for deep-rooted trees and applies water at appropriate intervals.

CHAPTER 6

CONCLUSION

All This system helps the farmer to choose the right crop by providing insights that ordinary farmers don't keep track of thereby decreasing the chances of crop failure and increasing productivity. It also prevents them from incurring losses. The system can be extended to the web and can be accessed by millions of farmers across the country. We could achieve an accuracy of 90 percent from the Decision Trees, an accuracy of 70.6 percent from the Support Vector Machine, an accuracy of 94.30 percent from the Logistic Regression and an accuracy of 99.09 percent from the Random Forest model. Further development is to integrate the crop recommendation system with another subsystem, yield predictor that would also provide the farmer an estimate of production if he plants the recommended crop.

FUTURE SCOPE

Future Scope of AI-Based Crop Monitoring System:

The AI-based crop monitoring system presents significant potential for further enhancement and expansion in the agricultural sector. The following future developments and applications can improve its performance, scalability, and impact:

Integration with IoT Devices:

The system can be integrated with IoT sensors to gather real-time environmental data, such as soil moisture, temperature, humidity, and pH levels. This will enable precise monitoring and automated decision-making, enhancing crop health management.

Predictive Analytics for Crop Yield:

By leveraging machine learning algorithms, the system can analyze historical data to predict crop yields accurately. This will help farmers in resource planning, Enhanced Drone Technology for Monitoring:

Advanced drone technology equipped with high-resolution cameras and AI algorithms can provide aerial imagery for large-scale crop monitoring. This can improve the detection of pest infestations, diseases, and nutrient deficiencies.

Integration with Weather Forecasting Systems:

Combining AI models with weather prediction data will help provide recommendations on optimal planting times, irrigation schedules, and preventive measures for extreme weather conditions.

REFERENCES

[1] Fahad Kamraan Syed, Agniswar Paul, Ajay Kumar, Jaideep Cherukuri. “Low-cost IOT+ML design for smart farming with multiple applications” 10th International Conference on Computing, Communication Networking Technologies, 2019.

[2] Archana Gupta, Dharmil Nagda, Pratiksha Nikhare, Atharva Sandbhor IEEE “ Smart Management of Crop Cultivation using IoT and MachineLearning”2019.

[3] Radhika, Narendiran, “Kind of Crops and Small Plants Prediction using IoT with Machine Learning,” International Journal of Computer & Mathematical Sciences, 2018.

[4] Anguraj.Ka, Thiyaneswaran.Bb, Megashree.Gc, Preetha Shri.J.Gd, Navya.Se, Jayanthi. Jf, “Crop Recommendation on Analyzing Soil Using Machine Learning” 2020.

[5] A. Mythili , “Classification of Soil and Crop Suggestion using Machine Learning Techniques”, IEEE 2019.

[6] Mehta, P., Shah, H., Kori, V., Vikani, V., Shukla, S., & Shenoy, M. “Survey of unsupervised machine learning algorithms on precision agricultural data”, IEEE 2018.

[7] Rani Holambe, Pooja Patil, Padmaja Pawar, Saurabh Salunkhe , Mr. Hrushikesh Joshi, “IOT based Crop Recommendation, Crop Disease Prediction and Its Solution” Rani Holambe, Pooja Patil, Padmaja Pawar, Saurabh Salunkhe , Mr. Hrushikesh Joshi, 2019 IRJET

[8] Tom M. Mitchell, Machine Learning, India Edition 2013, McGrawHill Education.

[9] https://data.gov.in/

[10] Kaggle: https://www.kaggle.com/datasets

APPENDIX

```
import pandas as pd #
Reading the data
crop_data_path=      ../Data-      raw/cpdata.csv'
fertilizer_data_path='../Data-raw/Fertilizer.csv'
crop=pd.read_csv(crop_data_path)
fert=pd.read_csv(fertilizer_data_path)
#Function      for      lowering      the      cases
defchange_case(i):
i = i.replace(" """) i = i.lower()
return

# using extract labelsl on crop to get all the data related to those labels
new_crop = pd.DataFrame(columns = crop.columns)

new_fert            =   pd.DataFrame(columns
                        fert.columns) for label in
extract_labels:

new_crop
new_crop.append(crop[crop['label']   label])  for label in
extract_labels:

new_fert                      new_fert.append(fert[fert['Crop']
label].iloc[0]) print(new_crop)

print(new_fert)
score          =    cross_val_score(DecisionTree,     features,
target,cv=5) print(score)
```

```
#Saving trained Decision Tree model
import pickle
# Dump the trained Naive Bayes classifier with Pickle
DT_pkl_filename = '../models/DecisionTree.pkl'
# Open the file to save as pkl file
DT_Model_pkl=
open(DT_pkl_filename,                'wb')
pickle.dump(DecisionTree,
DT_Model_pkl)
# Close the pickle instances
DT_Model_pkl.close()
from sklearn.ensemble import RandomForestClassifier


RF          =       RandomForestClassifier(n_estimators=20,
random_state=0) RF.fit(Xtrain,Ytrain)
predicted_values = RF.predict(Xtest)

x          =       metrics.accuracy_score(Ytest,
predicted_values) acc.append(x)
model.append('RF') print("RF's
Accuracy is: ", x)
print(classification_report(Ytest,predicte
d_values)) # Cross validation score (Random
Forest)
score
cross_val_score(RF,features,target,cv=5)
print(score)
import pickle
# Dump the trained Naive Bayes classifier with Pickle
RF_pkl_filename = '../models/RandomForest.pkl'
#Open   the   file   to   save   as   pkl   file
RF_Model_pkl   =   open(RF_pkl_filename,
'wb') pickle.dump(RF, RF_Model_pkl)
#Close  the        pickle
instances RF_Model_pkl.close()
score=cross_val_score(DecisionTree,features,
target,cv=5) print(score)
#Saving trained Decision Tree model
import pickle
# Dump the trained Naive Bayes classifier with Pickle
DT_pkl_filename = '../models/DecisionTree.pkl'
```

```
# Open the file to save as pkl file
DT_Model_pkl
open(DT_pkl_filename,                'wb')
pickle.dump(DecisionTree,
DT_Model_pkl)
# Close the pickle instances
DT_Model_pkl.close()
```