International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 2, 2026

Smart Farming

Ms J Lavanya[1],D.Nikitha[2], K. Anusha[3], K. Yamini[4]
[1]Assistant Professor; Department Of Computer Science And Engineering(AI&Ml) Bhoj Reddy Engineering College For Women Hyderabad India
[2,3,4]B.Tech Students; Department Of Computer Science And Engineering(AI&Ml) Bhoj Reddy Engineering College For Women Hyderabad India
Mail Id; dillinikitha625@gmail.com[2], kurlemanusha@gmail.com[3],Kunchalayamini06@gmail.com[4]

Abstract

Agriculture is a fundamental contributor to economic development, yet farmers continue to face significant challenges in making accurate decisions related to crop selection, fertilizer application, and plant disease control. To address these issues, this project proposes a Smart Agriculture System using Machine Learning, designed as an intelligent mobile-based solution that supports farmers in improving agricultural productivity through data-driven insights.The system employs machine learning techniques to recommend suitable crops based on key soil and environmental parameters such as nitrogen, phosphorus, potassium levels, soil pH, rainfall, and climatic conditions. In addition, it provides fertilizer recommendations tailored to soil nutrient deficiencies and incorporates plant disease detection using image processing and classification algorithms.To further enhance usability, the system integrates an AI-powered chatbot that enables farmers to interact in natural language and obtain instant responses to agriculture-related queries. The backend is implemented using Flask, while the frontend is developed using React Native to ensure cross-platform mobile accessibility. Communication between modules is handled through RESTful APIs, and the application can be easily accessed via QR code using Expo integration.This proposed solution aims to bridge the gap between modern artificial intelligence technologies and traditional farming practices. By providing real-time recommendations and intelligent assistance, the system helps improve crop yield, optimize resource utilization, and promote sustainable agricultural practices.

Keywords

Smart Agriculture, Machine Learning, Crop Recommendation, Fertilizer Prediction, Plant Disease Detection, Artificial Intelligence, Mobile Application, Precision Farming, Deep Learning, Sustainable Agriculture

Introduction

Agriculture is a crucial sector that sustains the livelihood of a large population, particularly in developing countries like India. Despite its importance, farmers continue to face several challenges such as unpredictable climatic conditions, improper selection of crops, lack of awareness regarding soil health, and difficulties in identifying plant diseases at early stages. These limitations often result in reduced agricultural productivity and financial instability for farmers. In recent years, the advancement of artificial intelligence and mobile computing has opened new opportunities to modernize traditional farming practices and make them more efficient and data-driven.The proposed project, *Smart Agriculture Using Machine Learning*, is designed to support farmers by providing intelligent recommendations and real-time assistance through a mobile-based platform. The system utilizes machine learning algorithms to analyze key soil and environmental parameters such as nitrogen, phosphorus, potassium levels, soil pH, rainfall, and climatic conditions in order to suggest suitable crops. It also provides fertilizer recommendations based on soil nutrient requirements and crop needs. Furthermore, the system incorporates a plant disease detection module that uses image processing techniques to identify crop diseases and recommend possible treatments.To enhance user interaction and accessibility, an AI-powered chatbot is integrated into the application, enabling farmers to obtain instant responses to agriculture-related queries in a simple and interactive manner. The system architecture is developed using Flask for backend processing and React Native for the frontend, ensuring smooth cross-platform mobile functionality. Communication between modules is handled through RESTful APIs, and the application is deployed in a user-friendly manner using QR code-based access via Expo. Overall, this system aims to bridge the gap between technology and agriculture by improving decision-making, increasing crop productivity, and promoting sustainable farming practices.

Existing System

In conventional agricultural practices, farmers primarily depend on traditional knowledge, past experience, and suggestions from local agricultural experts for making critical farming decisions. Activities such as crop selection, fertilizer application, and disease identification are often carried out based on intuition rather than scientific analysis. Information is usually obtained through informal communication channels such as local markets, peer farmers, or agricultural officers, which may not always be accurate, timely, or reliable.Although several digital agricultural tools and platforms exist, many of them have limited capabilities. Most systems provide only basic advisory services and often lack advanced features such as real-time interaction, image-based disease detection, and integrated decision-making support. In addition, some applications are restricted by language barriers or do not combine multiple functionalities into a unified platform, making them less effective for comprehensive agricultural assistance.

Proposed System

The proposed system, *Smart Agriculture Using Machine Learning*, introduces an intelligent and integrated solution designed to enhance agricultural decision-making. It combines machine learning, image processing, and artificial intelligence technologies within a single mobile application to assist farmers in various aspects of farming. The system analyzes soil and environmental attributes such as nutrient content, pH level, rainfall, and temperature to recommend suitable crops. It also provides fertilizer suggestions tailored to specific soil conditions and crop requirements.In addition, the system features a plant disease detection module that allows users to upload images of affected crops for automatic identification and treatment suggestions using image classification techniques. A key enhancement of the proposed solution is the inclusion of an AI-based chatbot, which offers real-time responses to user queries and supports farmers with easy-to-understand agricultural guidance. The application is developed using Flask for backend services and React Native for frontend mobile interaction, ensuring cross-platform compatibility. API-based communication enables smooth data exchange between components, and QR code integration allows simple access through mobile devices. This system ultimately aims to improve agricultural productivity, reduce dependency on manual decision-making, and promote smart and sustainable farming practices.

Literature Survey

Recent developments in agriculture have increasingly focused on the integration of Internet of Things (IoT), machine learning, and data-driven approaches to enhance farming productivity and decision-making. Researchers have proposed various intelligent systems aimed at supporting farmers through real-time monitoring, prediction, and recommendation services.Syed et al. introduced a cost-effective smart farming framework that combines IoT sensors with machine learning techniques. The system collects environmental parameters such as temperature, humidity, and soil moisture and processes them using predictive models to support irrigation scheduling and crop management. Although the solution is economical and suitable for small-scale farmers, it faces limitations in terms of scalability and advanced analytical capabilities.Gupta et al. developed a smart agricultural management system that integrates IoT devices with predictive analytics for real-time decision support. The system continuously monitors field conditions and recommends suitable crops and optimized resource utilization strategies. While it improves efficiency and reduces manual intervention, its performance is highly dependent on the accuracy and reliability of sensor data, which may limit its effectiveness in large-scale environments.Radhika proposed a predictive framework for estimating crop yield using machine learning and IoT-based data acquisition. The study utilizes both historical and real-time agricultural data to forecast crop performance and assist farmers in planning cultivation strategies. However, the effectiveness of the model is highly dependent on the availability of large and well-structured datasets.Angurai et al. presented a crop recommendation system based on soil parameter analysis using machine learning algorithms. The system evaluates soil characteristics such as nutrient content, pH level, and moisture to suggest suitable crops for cultivation. This approach enhances soil utilization efficiency, although it requires periodic updates of soil data to maintain accuracy.Mythili examined different machine learning classification techniques for soil analysis and crop prediction. The study compares models such as decision trees and support vector machines, concluding that algorithm selection significantly influences prediction accuracy. However, computational complexity may pose challenges for real-time implementation.Mehta et al. conducted a comprehensive review of unsupervised learning techniques in agricultural data analysis. The study highlights clustering methods used to identify hidden patterns in crop and soil datasets. While these techniques help in discovering useful insights, the absence of labeled data can limit result interpretability.Holambe et al.proposed an integrated agricultural system combining crop recommendation and plant disease detection using IoT and machine learning. The system uses image processing techniques for disease identification and also provides crop suggestions based on environmental conditions. Although the approach is comprehensive, it requires significant computational resources for real-time processing.

Requirement Analysis

Functional Requirements

The proposed system is designed to provide intelligent agricultural support through multiple functional modules. It allows users to register and securely log into the system. The application provides crop recommendation based on soil parameters such as nitrogen, phosphorus, potassium, pH level, and rainfall. It also suggests suitable fertilizers based on soil conditions and selected crops. Additionally, the system includes a plant disease detection module that enables users to upload crop images for automatic disease identification and treatment suggestions. An AI-powered chatbot is integrated to respond to agriculture-related queries in real time, and the system displays results and recommendations in a user-friendly format.

Non-Functional Requirements

The system is designed with usability and performance in mind. It provides a simple and intuitive interface to ensure ease of use for farmers with minimal technical knowledge. The application is optimized for fast response times in prediction and chatbot interactions. It ensures reliability and high availability for continuous usage. The system is scalable to support multiple users simultaneously and incorporates security mechanisms for user authentication and data protection. It is compatible with mobile devices and designed for maintainability to allow future enhancements and feature upgrades.

Computational Resource Requirements

Hardware Requirements

The system requires a minimum configuration of an Intel i5 processor or higher to ensure smooth processing. At least 8 GB of RAM is recommended for efficient execution of machine learning models. A minimum of 50 GB storage space is required to store datasets, models, and application files. A stable internet connection is necessary for API communication, chatbot functionality, and real-time data exchange.

Software Requirements

The system operates on Windows, Linux, or macOS operating systems. It is developed using Python and JavaScript programming languages. The backend is implemented using Flask, while the frontend is built using React Native with Expo. Machine learning functionalities are developed using Scikit-learn, along with libraries such as NumPy, Pandas, and OpenCV. The database used is SQLite or MySQL for storing application data. Development tools include Visual Studio Code, Jupyter Notebook, Android Studio, and GitHub. The system is currently deployed in a local environment with potential for future cloud integration.

3.4 Life Cycle Model

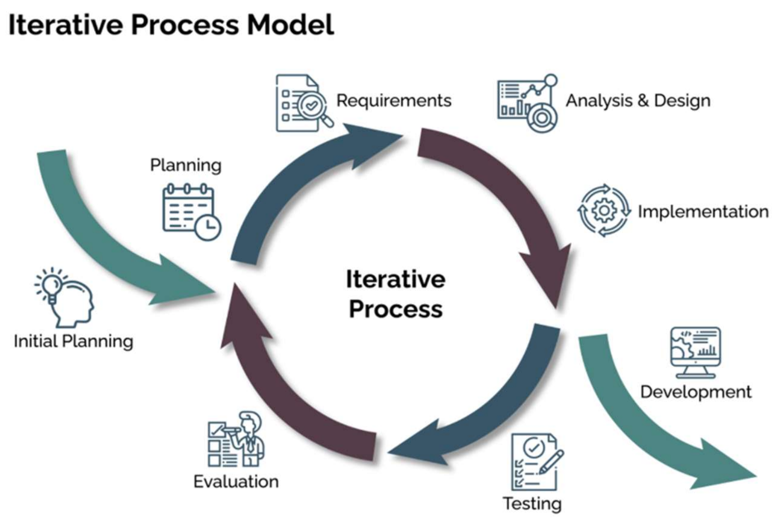

Fig: 1 Iterative Process Model

The proposed system follows the Iterative Process Model for software development. In this approach, the entire system is developed in repeated cycles, where each iteration includes phases such as requirement analysis, design, implementation, testing, and evaluation. Instead of delivering the complete system at once, the project is divided into smaller functional modules that are developed and improved incrementally.After each iteration, the developed components are reviewed, and feedback is incorporated to enhance system performance and functionality. This iterative refinement continues until the final system is fully developed and optimized. The model is particularly beneficial as it allows early detection of errors, supports flexibility in requirement changes, and ensures continuous improvement of the system throughout the development lifecycle.

Design

The design of the proposed *Smart Agriculture Using Machine Learning* system describes the overall structure of the application, the interaction between its components, and the flow of data throughout the system. The architecture of the system defines how different modules collaborate to process user requests and generate meaningful outputs such as crop recommendations, fertilizer suggestions, disease detection results, and chatbot responses. The system architecture is broadly categorized into software architecture and technical architecture, each playing a crucial role in ensuring efficient system performance and modular development.

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 2, 2026

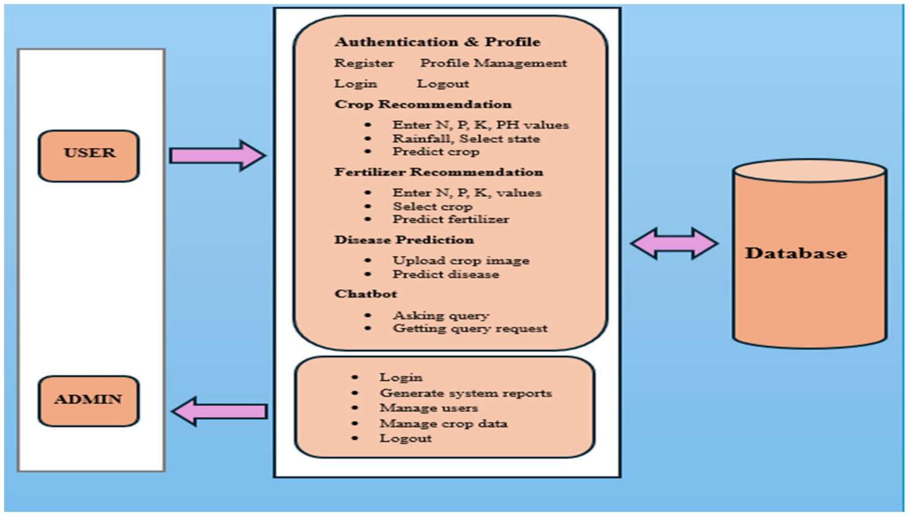

Fig 4.1: Software Architecture

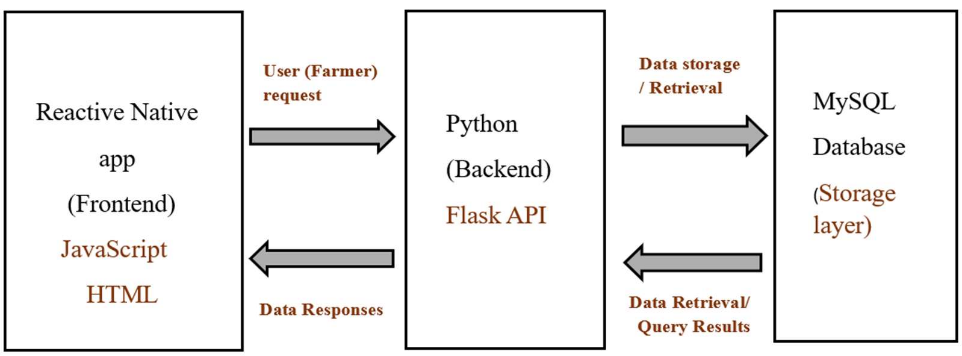

Fig 4.2: Technical Architecture

The software architecture focuses on the logical arrangement of system components, including the mobile application, backend server, machine learning models, and database. It illustrates how user inputs from the mobile interface are transmitted to the backend system, processed by machine learning algorithms, and returned as predictions to the user. The technical architecture, on the other hand, represents the underlying technologies used in system development, such as Flask for backend services, React Native for frontend development, and Scikit-learn and deep learning frameworks for model implementation. Together, these architectures ensure smooth integration and efficient communication between all system modules.The system behavior is further illustrated using UML diagrams, which include use case diagrams, class diagrams, sequence diagrams, activity diagrams, communication diagrams, and data flow diagrams. The use case diagrams define the interaction between actors such as users and the system, where users can perform actions like crop prediction, fertilizer recommendation, and disease detection. The class diagram represents the structural relationship between different classes and their attributes and methods, helping in understanding system design at a detailed level.Sequence diagrams describe the chronological flow of messages between components during execution, showing how requests are processed step by step. Activity diagrams illustrate the workflow of the system, including decision-making processes and parallel operations for both user and admin modules. Communication diagrams focus on object interactions and message exchanges within the system. The data flow diagram represents how data moves between different components, including input collection, processing through machine learning models, and output generation.

Implementation

The implementation of the Smart Agriculture system is carried out using a combination of machine learning, deep learning, and mobile application development technologies. The system is designed to provide intelligent agricultural support through multiple functional modules,

ISSN 2347–3657

Volume 14, Issue 2, 2026

including crop recommendation, fertilizer prediction, plant disease detection, and an AI-based chatbot.The development environment is set up using Python as the primary programming language, along with frameworks such as Flask for backend development and React Native for mobile application development. Additional libraries including Scikit-learn, PyTorch, NumPy, Pandas, and OpenCV are used for implementing machine learning and image processing functionalities. Tools such as Visual Studio Code and Jupyter Notebook are used for coding, model training, and testing.Agricultural datasets containing parameters like nitrogen, phosphorus, potassium, temperature, humidity, pH, and rainfall are collected and preprocessed to remove inconsistencies and prepare them for model training. Machine learning models such as Random Forest are trained for crop and fertilizer recommendation tasks, while a Convolutional Neural Network (CNN), specifically ResNet-based architecture, is used for plant disease classification using leaf images.The backend of the system is developed using Flask, which handles API requests from the mobile application, processes inputs using trained models, and returns prediction results. The frontend mobile application is developed using React Native, providing a user-friendly interface for farmers to input data, upload images, and receive recommendations. A SQLite database is used to store user information and historical prediction data, ensuring efficient data management and retrieval.Finally, all system components, including the mobile application, backend server, machine learning models, and database, are integrated into a unified platform. This integration enables seamless communication between modules and ensures real-time prediction and recommendation services for end users.The system also utilizes various technologies such as Flask-Login, Flask-Bcrypt, and WTForms for authentication and form handling, while APIs like OpenWeatherMap are used to fetch external weather data. The chatbot functionality is implemented using a natural language processing-based service to provide real-time responses to user queries. Model persistence is achieved using pickle files, allowing trained models to be reused efficiently during deployment.

6. Testing

Testing is a critical phase in the development of the Smart Agriculture system, ensuring that all modules function correctly and deliver accurate results. The primary objective of testing is to validate system performance, identify errors, and ensure that the application meets functional and non-functional requirements. The system is tested for crop prediction accuracy, fertilizer recommendation correctness, disease detection reliability, and chatbot responsiveness.Different levels of testing are performed to ensure system quality. Unit testing is conducted to verify individual modules such as crop prediction, fertilizer recommendation, and disease detection independently. Integration testing ensures that different components, including the frontend, backend, and machine learning models, work together correctly and exchange data without errors. Functional testing validates whether the system outputs match the expected results based on given inputs, ensuring correct behavior as per requirements. System testing evaluates the complete application in a real-world environment to ensure overall reliability, performance, and usability.The system is also evaluated using black box and white box testing techniques. In black box testing, the system is tested based on input-output behavior without considering internal code structure, such as verifying crop prediction results from soil inputs. In white box testing, the internal logic of algorithms and model processing is examined to ensure correctness of implementation and decision-making flow.A set of test cases is designed to validate different functionalities of the system. These include login authentication, crop prediction with valid and invalid inputs, fertilizer recommendation based on nutrient levels, plant disease detection using both healthy and infected images, chatbot responses for various queries, and handling of incorrect or missing inputs. The system successfully handles all test cases, demonstrating reliable performance and accurate prediction capabilities across all modules.

Input And Output

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 2, 2026

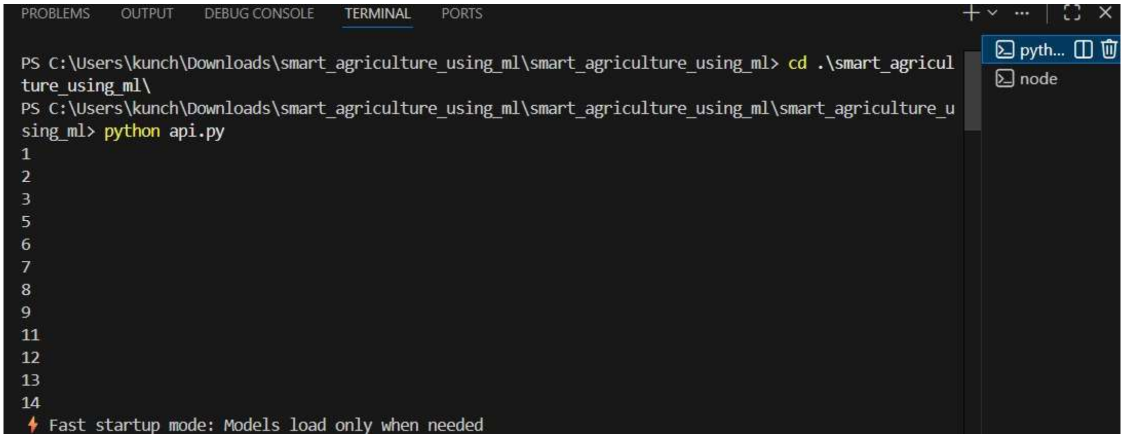

Fig: 1: backend

Fig 2: Login page

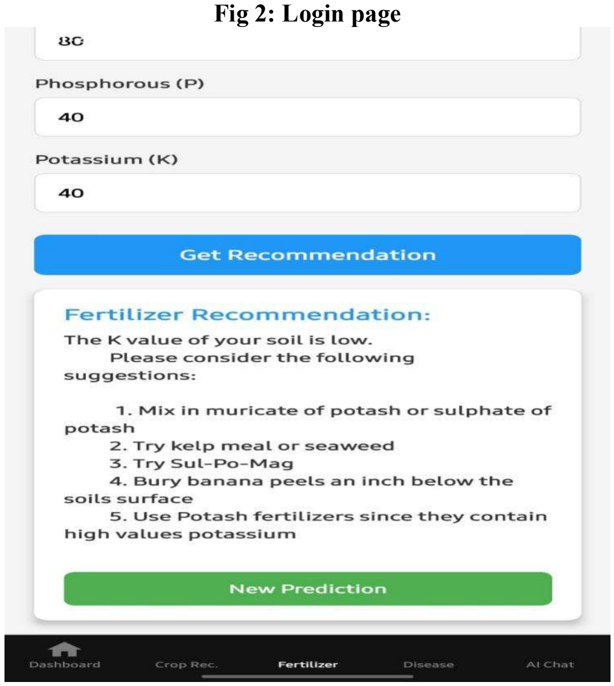

Fig 3: Fertilizer Recommendation

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 2, 2026

Fig 4: Disease Detection

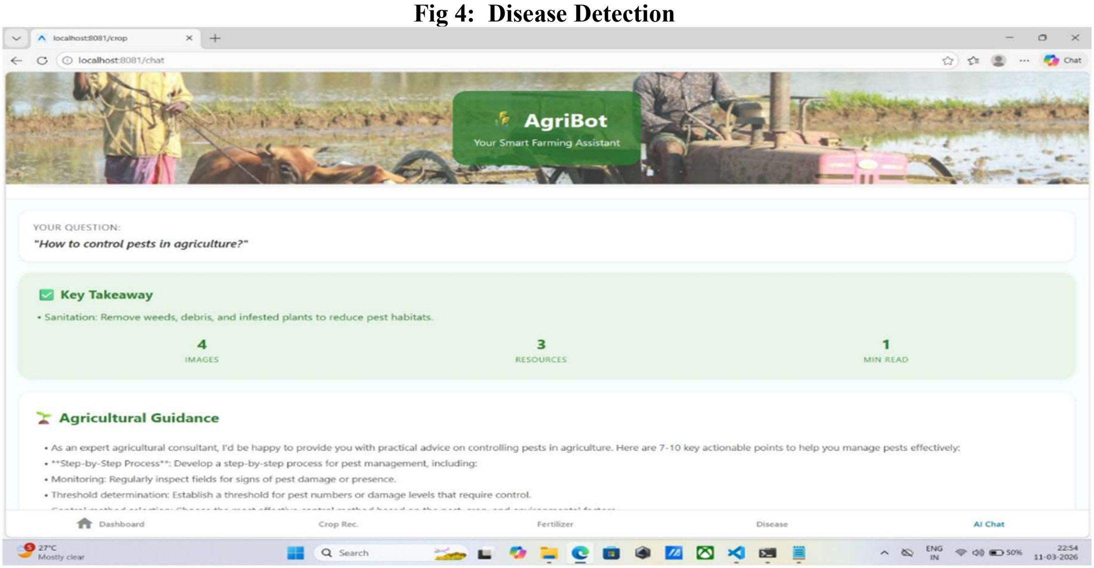

Fig 5: Chatbot Assistance

Conclusion

The proposed *Smart Agriculture Using Machine Learning* system effectively demonstrates the application of modern technologies such as machine learning, deep learning, and mobile application development in enhancing agricultural decision-making. The system provides intelligent support for farmers by offering accurate recommendations for crop selection and fertilizer usage based on soil and environmental parameters. By utilizing trained machine learning models, the system analyzes critical agricultural data to assist farmers in selecting suitable crops, thereby minimizing the risk of crop failure and improving productivity.In addition, the plant disease detection module enables early identification of crop diseases through image-based analysis, allowing timely preventive measures and reducing potential agricultural losses. The integration of an intuitive mobile application ensures that farmers can easily access recommendations and system functionalities in real time without requiring advanced technical knowledge. Overall, the system contributes significantly toward improving agricultural efficiency, reducing resource wastage, and promoting sustainable and data-driven farming practices.

Future Scope

The proposed system can be further enhanced by incorporating advanced technologies and extended functionalities. In the future, drone-based monitoring systems can be integrated to capture real-time crop health data over large agricultural fields. The inclusion of IoT-based sensor networks can provide continuous monitoring of soil conditions such as moisture, temperature, and nutrient levels, improving prediction accuracy. Additionally, satellite imagery analysis can be used for large-scale crop health assessment and yield estimation.From a user experience perspective, the system can be improved by adding multilingual support to assist farmers from diverse linguistic backgrounds. Offline functionality can be introduced to enable usage in areas with limited internet connectivity. Voice-based interaction systems and augmented reality-based disease detection tools can further enhance accessibility and usability.From a business intelligence perspective, future enhancements may include farm management analytics dashboards, crop market price prediction models, and supply chain integration features. Sustainability tracking systems can also be implemented to monitor environmental impact and promote eco-friendly farming practices.

9. References

1. F. K. Syed, A. Paul, A. Kumar, and J. Cherukuri, "Low-cost IoT and machine learning design for smart farming with multiple applications," *Proc. 10th Int. Conf. Computing, Communication and Networking Technologies*, 2019.
2. A. Gupta, D. Nagda, P. Nikhare, and A. Sandbhor, "Smart management of crop cultivation using IoT and machine learning," *IEEE Conference Proceedings*, 2020.
3. N. Radhika, "Crop and plant growth prediction using IoT with machine learning," *International Journal of Computer & Mathematical Sciences*, 2018.
4. K. Angurai, B. Thiyaneswaran, G. Megashree, J. Preetha Shri, S. Navya, and J. Jayanthi, "Crop recommendation using soil analysis and machine learning," *IEEE Conference Paper*, 2020.
5. A. Mythili, "Classification of soil and crop suggestion using machine learning techniques," *IEEE Proceedings*, 2019.
6. P. Mehta, H. Shah, V. Kori, V. Vikani, S. Shukla, and M. Shenoy, "Survey of unsupervised machine learning algorithms in precision agriculture," *IEEE Conference*, 2018.
7. R. Holambe, P. Patil, P. Pawar, S. Salunkhe, and H. Joshi, "IoT-based crop recommendation and disease prediction system," *IRJET*, 2019.
8. S. Mohanty, D. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, 2016.
9. A. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, Elsevier, 2018.
10. R. Sujatha and P. Isakki Raj, "Random forest-based crop recommendation system using soil and climate data," *IEEE International Conference on Intelligent Computing Systems*, 2020.