IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A Predictive System For Precision Agriculture: Crop, Disease And Fertilizer Prediction Using Machine Learning

[1]Priya N V, [2]Atharv Amit Gangrade,[3]Hari Priya L, [4]Ishwari Ratre, [5]Karan Chauhan

[1]Assistant Professor, [2]Student, [3]Student,[4]Student,[5]Student

Information Science and Engineering,

[1]Bangalore Institute of Technology, Bangalore, India

Abstract: The rapid advancement of technology, particularly machine learning (ML) and the Internet of Things (IoT), is revolutionizing agriculture by optimizing crop production and ensuring sustainability. This paper presents a web-based platform developed to assist farmers in selecting suitable crops, predicting fertilizer requirements, and diagnosing plant diseases. The platform uses data from Kaggle, ML models implemented in Flask, and ReactJS for the frontend. Farmers can input NPK values and city names to receive insights on temperature, humidity, and fertilizer requirements, while also benefiting from disease prediction through image uploads and SMS alerts. The system aims to improve farming practices, promote sustainability, and enhance productivity [3], [4].

***Index Terms* -** Machine Learning, Precision Agriculture, ReactJS, Flask, Crop Prediction, Disease Detection, SMS Alerts, Fertilizer Recommendation

I. INTRODUCTION

Agriculture has always been the foundation of human civilization, not only providing essential resources such as food and raw materials but also playing a critical role in the global economy. According to the United Nations Food and Agriculture Organization (FAO), agriculture contributes approximately 4% to the global GDP and employs nearly 27% of the global workforce. However, the agricultural sector faces mounting challenges that threaten its sustainability and efficiency. These challenges include rapid population growth, climate change, resource scarcity, and unpredictable weather patterns, all of which necessitate a paradigm shift in farming practices [1].

With the global population expected to exceed 9.7 billion by 2050, food production must increase by approximately 70% to meet the growing demand. Traditional farming methods, while effective in the past, are becoming insufficient to handle these pressures due to their reliance on intensive resource consumption and lack of precision. The need to increase productivity, optimize resource usage, and reduce environmental impacts has led to the rise of **precision agriculture**, also known as Agriculture 4.0, which integrates advanced technologies such as **machine learning (ML)**, **the Internet of Things (IoT)**, and **big data analytics** into farming practices [2], [3].

Precision agriculture is characterized by its ability to make data-driven decisions, enabling farmers to improve yield predictions, optimize resource allocation, and detect diseases early. For example, **IoT-enabled sensors** can monitor soil conditions, temperature, and humidity, while ML algorithms analyze this data to provide actionable insights [3]. These innovations reduce waste, improve productivity, and help farmers adapt to changing environmental conditions. However, despite its potential, the adoption of precision agriculture has been limited, especially in developing countries, due to high implementation costs, lack of technical expertise, and challenges in integrating diverse datasets [4].

To address these challenges, this paper presents a **web-based precision agriculture platform** that empowers farmers to make informed decisions regarding crop selection, fertilizer usage, and disease management. By integrating ML models with an intuitive interface built on ReactJS and Flask, the system enables farmers to input soil parameters such as NPK (Nitrogen, Phosphorus, Potassium) values and location details to receive tailored recommendations. The system also incorporates a **disease detection feature**, allowing farmers to upload images of unhealthy plants for analysis using a pre-trained convolutional neural network (CNN) [3], [5]. Additionally, an **SMS-based alert service** provides farmers with daily weather updates and critical notifications, ensuring they remain informed about environmental conditions [4].

The importance of such a system cannot be overstated. The unpredictability of weather due to climate change, combined with resource scarcity, makes farming increasingly risky. Many farmers, especially smallholders, often lack the tools and information needed to make strategic decisions about their crops. This can result in overuse of fertilizers, underuse of irrigation, and significant crop losses due to undiagnosed plant diseases. By addressing these gaps, this platform aims to reduce the knowledge disparity among farmers and promote sustainable agricultural practices [1], [2].

This paper is organized as follows: Section 2 reviews related work on the application of ML and IoT in agriculture, highlighting gaps in existing systems. Section 3 describes the methodology and development of the proposed platform, including its frontend, backend, and ML models. Section 4 discusses the system's architecture and implementation details. Section 5 presents the results and evaluates the platform's performance. Section 6 discusses the implications, challenges, and future improvements. Finally, Section 7 concludes with the system's contributions to advancing precision agriculture.

1.1 Related Work

Several studies have demonstrated the transformative potential of ML and IoT in agriculture:

- **Crop Selection and Management**: Research by Tseng et al. highlights the use of IoT and ML to monitor farm conditions and recommend suitable crops based on environmental data [1]. Similarly, Godara et al. introduced a query-response system that assists farmers by analyzing historical call data to provide solutions for plant protection [2].
- **Disease Detection**: Disease identification using ML models has proven effective in diagnosing plant diseases through image recognition. Sharma et al. reviewed the use of CNNs and computer vision for detecting diseases and classifying crop conditions [3] Precision Irrigation and Fertilizer.

Whildies provide valuable insights, few systems integrate multiple functionalities, such as crop prediction, disease detection, and fertilizer recommendation, into a unified platform tailored for farmers in diverse regions.

II. PROPOSED SYSTEM

The proposed system is a multi-functional web-based platform designed to support farmers by providing actionable insights to improve agricultural productivity. It integrates machine learning (ML) models, real-time weather updates, and user-friendly interfaces to address key challenges in precision agriculture. By leveraging environmental and soil data, the system enables farmers to make data-driven decisions on crop selection, fertilizer application, and plant disease management [4].

2.1 SYSTEM OBJECTIVES

- **Crop Recommendation:** Suggest the most suitable crops for cultivation based on soil nutrient values (NPK), temperature, and humidity [3].
- **Fertilizer Recommendation:** Provide optimized fertilizer recommendations to improve soil health and crop yield while minimizing environmental impact [1].

- **Disease Prediction:** Identify plant diseases from uploaded images of unhealthy crops, offering specific management advice [5].
- **Real-Time Weather Alerts:** Enable farmers to subscribe to daily weather updates and receive SMS alerts for significant weather events [2].

2.2 WORKFLOW

The workflow of the proposed system has been carefully designed to ensure seamless user interaction and efficient processing of data, enabling farmers to receive actionable insights with ease and accuracy. The process begins with the **Input Phase**, where farmers interact with the platform by providing essential information. This includes entering soil parameters such as nitrogen (N), phosphorus (P), and potassium (K) values, along with their city name to identify location-specific environmental conditions [1]. Additionally, the system allows farmers to upload images of unhealthy crops for disease diagnosis. An optional feature Penables users to subscribe to SMS notifications, ensuring they receive daily weather updates and important alerts directly on their mobile devices.

Once the input data is provided, it transitions to the **Processing Phase**, where the backend server plays a critical role. The server first validates the input data to ensure correctness and completeness. For instance, NPK values must fall within acceptable agricultural ranges, and uploaded images are checked for compatibility with the disease detection model. Following validation, the system selects and executes the appropriate machine learning model based on the farmer's requirements. These models include crop prediction, fertilizer optimization, and disease detection. To enhance the accuracy and relevance of predictions, real-time weather data is retrieved from external APIs. This dynamic data integration ensures that the system considers current environmental conditions, such as temperature and humidity, when generating recommendations [3].

Finally, in the **Output Phase**, the system delivers the results back to the farmer in an easily interpretable format. Crop recommendations, fertilizer suggestions, and disease diagnoses are displayed on an interactive web dashboard, providing detailed insights for informed decision-making. For users who have subscribed to the SMS service, the system sends concise alerts containing weather updates or critical warnings, such as anticipated storms or temperature fluctuations. This ensures that farmers are equipped with timely information, regardless of internet accessibility. By streamlining the input, processing, and output phases, the workflow ensures that the system operates efficiently, enabling farmers to address agricultural challenges with confidence and precision [5].

2.3 FUNCTIONAL MODULES

The system is organized into distinct functional modules, each designed to manage specific aspects of the workflow and ensure a smooth and efficient user experience. These modules collectively contribute to the overall functionality and reliability of the platform.

The **Input Module** serves as the entry point for farmers to interact with the system. This module provides a user-friendly interface where farmers can input critical soil parameters, including nitrogen (N), phosphorus (P), and potassium (K) values, along with their location, specified by their city name. These inputs help the system tailor its recommendations to local environmental and soil conditions. Additionally, the module includes an image upload feature specifically designed for plant disease diagnosis. Farmers can upload images of unhealthy crops, which the system processes using advanced machine learning algorithms to identify potential diseases. To further enhance accessibility, the module also offers a subscription option for SMS alerts. This feature allows farmers to receive daily weather updates and critical warnings directly to their mobile phones, ensuring they stay informed even in areas with limited internet connectivity.

Once the data is collected, it is passed to the **Processing Module**, which handles the critical task of validating and routing the input data. This module ensures that all inputs meet predefined criteria, such as acceptable ranges for NPK values and image formats compatible with the disease detection model. After validation, the module determines the appropriate machine learning model to execute based on the farmer's specific query, whether it involves crop prediction, fertilizer optimization, or disease diagnosis. The module also integrates real-time weather data from external APIs to refine predictions and enhance the system's accuracy. This dynamic incorporation of environmental factors, such as temperature and humidity, ensures that recommendations are both precise and contextually relevant.

The **Prediction Module** is where the core intelligence of the system is executed. This module employs advanced machine learning algorithms to analyze the validated input data and generate actionable outputs. For example, the crop prediction model recommends the most suitable crops based on soil and environmental conditions, while the fertilizer recommendation model provides optimal fertilizer combinations to improve crop yields sustainably. For disease diagnosis, the module uses a convolutional neural network (CNN) to classify uploaded plant images and provide detailed insights into the detected diseases, including recommended treatments. Once the predictions are generated, the module sends the results back to the farmer. Outputs are displayed on the web dashboard in a visually intuitive format, and for subscribed users, concise SMS alerts are delivered directly to their mobile phones.

Together, these functional modules form a cohesive system that addresses the diverse needs of farmers. The integration of input validation, real-time data processing, and advanced predictive analytics ensures that the platform delivers reliable and actionable insights to support informed agricultural decision-making.

2.4 TECHNOLOGY STACK

2.4.1 FRONTEND TECHNOLOGY: REACTJS

The frontend of the system is built using **ReactJS**, a widely used JavaScript library for building dynamic and interactive user interfaces. ReactJS enables the creation of reusable UI components, ensuring a responsive and modular design.

Features Implemented:

- Data Input Forms: Allows users to enter NPK values, upload crop images, and subscribe to SMS services.
- Prediction Dashboard: Displays crop recommendations, fertilizer suggestions, and disease diagnosis results in a clean and intuitive layout.
- Real-Time Weather Display: Fetches and updates current weather information for the farmer's location dynamically.
- Responsive Design: Optimized for use on both desktop and mobile devices to cater to farmers with varying access to technology.

2.4. BACKEND TECHNOLOGY: FLASK

The backend of the system is developed using **Flask**, a lightweight Python web framework known for its simplicity and flexibility. Flask provides the core infrastructure for handling HTTP requests, processing user inputs, and communicating with machine learning models.

Key Responsibilities:

- API development to enable seamless interaction between the frontend and backend.
- Data validation to ensure correctness and reliability of inputs.
- Integration of machine learning models for predictions and recommendations.
- Handling external API calls for fetching real-time weather data.

Flask’s minimalistic structure and modular nature make it an excellent choice for building scalable applications like this platform.

2.4.3 PYTHON LIBRARIES AND THEIR ROLES

Python serves as the backbone of the system's predictive capabilities, and several libraries have been utilized for specific functionalities:

1. Machine Learning and Data Processing

- **Scikit-learn**: Used for developing the crop prediction and fertilizer recommendation models. Scikit-learn is a robust library for machine learning, offering tools for classification, regression, clustering, and data preprocessing.

  Role in the System:

  - Decision Tree Classifier: For predicting the most suitable crops based on NPK values and environmental factors.
  - Regression Models: For fertilizer recommendations, predicting optimal quantities based on soil nutrient levels.

- **NumPy**: A fundamental library for numerical computation in Python. It is used for handling arrays, performing matrix operations, and preprocessing data for machine learning models.

  Role in the System:

  - Efficient handling of large datasets for NPK values and crop yield data.
  - Support for model training by providing optimized numerical operations.

- **Pandas**:Provides high-performance data manipulation and analysis tools. It is used extensively for cleaning and preprocessing the Kaggle dataset.

  Role in the System:

  - Data wrangling: Cleaning and organizing input data for machine learning.
  - Analysis: Generating insights from historical crop and fertilizer data

2.4.4 DEEP LEARNING FOR DISEASE DETECTION

- **TensorFlow**: TensorFlow, along with its high-level API Keras, is used to develop and train the convolutional neural network (CNN) for disease detection. These frameworks provide tools for building, training, and deploying deep learning models.

  Role in the System:

  - Image Processing: The CNN model processes uploaded images of unhealthy crops to identify diseases.
  - Model Deployment: Trained models are integrated with the Flask backend for real-time disease prediction.

III. SYSTEM ARCHITECTURE

The system architecture defines the structure, components, and interactions within the proposed platform. It follows a client-server model, ensuring modularity, scalability, and efficient communication between the frontend, backend, database, and external services. This section details the architecture and includes visual representations of the system's functionality [3], [5].

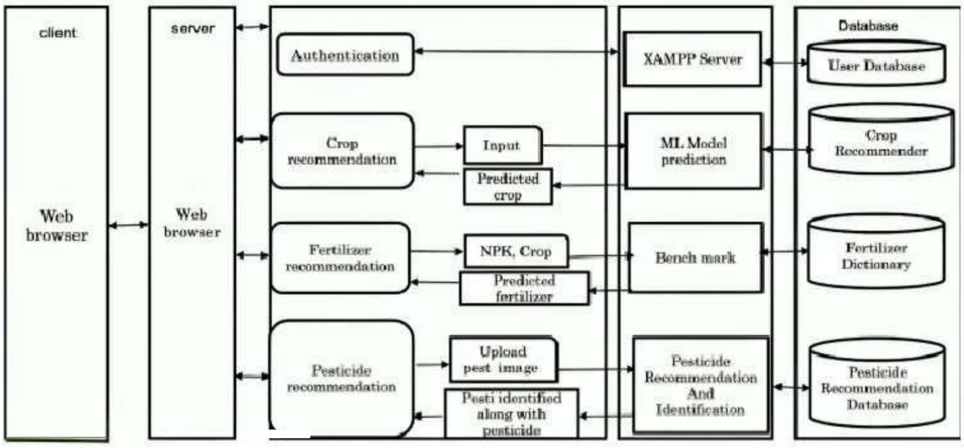

Figure 1: System Architecture Diagram This figure illustrates the overall system architecture, highlighting the interaction between the frontend, backend, machine learning models, database, and external APIs. It depicts the client-server model employed to ensure modularity, scalability, and seamless data flow. The architecture also showcases how real-time data integration and predictive analytics support farmers in decision-making.

3.1 DATAFLOW DIAGRAM

The **Data Flow Process** for the system follows a structured and logical progression, integrating input data, refining it, and generating predictions to assist farmers effectively. The workflow begins with the **Data Integration** block, which consolidates inputs from multiple sources, including user- provided information (such as NPK values and plant images), environmental data retrieved from external APIs, and historical datasets. This integrated data is then forwarded to the **Attribute Selection** block, where only the most relevant features are selected for further processing, ensuring that the subsequent stages handle only meaningful data.

Next, the data passes through the **Preprocessing** block, which prepares it for analysis. During this phase, steps such as handling missing values, data normalization, and data transformation are performed. The preprocessed data is then divided into two parallel paths. One path sends the data to the **Elimination** block, where redundant or irrelevant attributes are removed to reduce noise and enhance efficiency. Simultaneously, the second path routes the data to the **Mean Algorithm** block, where mean values are calculated to address missing data points and normalize attribute values. These processes ensure the dataset is refined and consistent [1], [3].

The outputs from the **Elimination** and **Mean Algorithm** blocks are then merged to create a **Cleaned Dataset** that is free of inconsistencies and noise. This cleaned dataset is further sent to the **Feature Selection Algorithm** block, where advanced techniques are applied to identify the most significant features that will contribute to the accuracy and reliability of the predictive models. The selected features are then integrated into a cohesive dataset in the **Integration** block, forming the foundation for modeling.

Once the dataset is prepared, it is sent to the **SVM Model** block, which leverages a Support Vector Machine algorithm to analyze the data and generate preliminary predictions. These outputs are further processed in the **Prediction Model** block, where final predictions are refined and optimized based on the selected features and the trained model's capabilities. Finally, the results are passed to the **Result Block**, where they are formatted and presented to the user via an intuitive dashboard or delivered via SMS alerts.

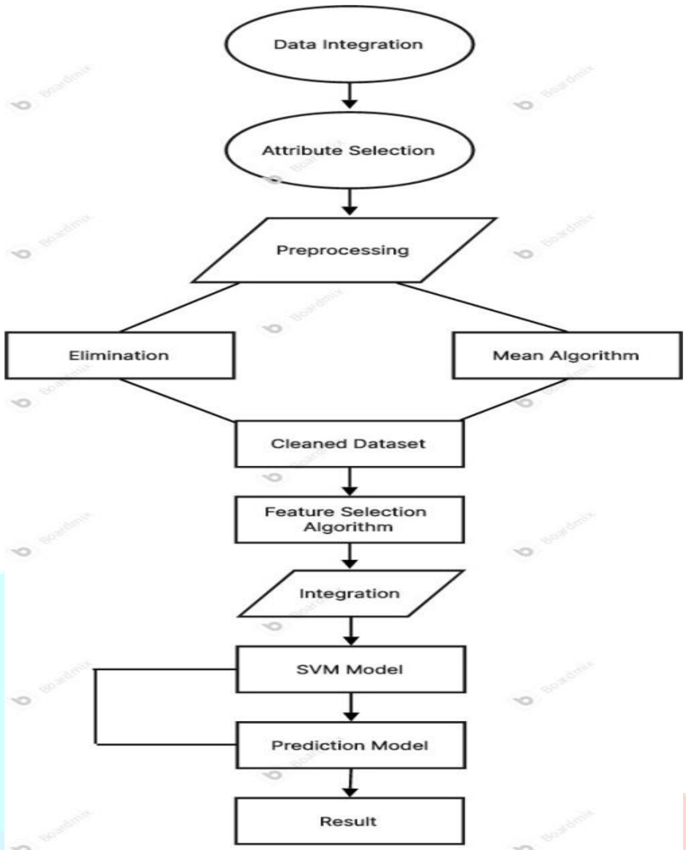

Figure 2: Data Flow Diagram This figure represents the structured flow of data within the system, starting from user inputs and external data sources to the final outputs delivered to farmers. Key stages include data integration, preprocessing, feature selection, predictive modeling, and result delivery. The diagram emphasizes the logical progression of data and ensures efficient processing for accurate recommendations.

This structured data flow ensures seamless processing, high accuracy, and actionable insights tailored to the needs of farmers.

3.2 USECASE DIAGRAM

The **Use Case Diagram** for the system revolves around the interaction between various actors, including farmers, databases, artificial intelligence (AI) systems, and agriculture officers, to support agricultural decision-making and query resolution. The primary actor in the system is the **Farmer**, who interacts with the platform to access agricultural recommendations, receive updates, and resolve queries. Supporting actors include the **Farmer_DB**, which stores farmer-specific information such as registration and authentication details; the **Data Base**, a central repository containing agricultural data like crop details, weather information, and productivity records; the **Agriculture Expert (AI)**, an intelligent system that provides data-driven insights and recommendations; and the **Agriculture Officer**, a human expert available for further assistance.

The system encompasses several use cases:

1. **Registration**: Farmers register with the system by providing their details, which are stored securely in the **Farmer_DB**. This ensures that each farmer has a unique account for accessing personalized services.

2. **Authentication**: To access the system's functionalities, farmers authenticate themselves by entering their credentials. The system verifies these credentials against the **Farmer_DB**, allowing authorized access only.

3. **Agri-Info**: Farmers can request detailed agricultural information, including sub-categories like crop details, weather updates, irrigation guidelines, fertilizer recommendations, productivity insights, and pest control measures. The system retrieves the requested data from the **Data Base** and presents it to the farmer.

4. **Diagnosis Agriculture Status**: Farmers can use the system to diagnose the current status of their agricultural operations. The system analyzes data from the **Data Base** using the **Agriculture Expert (AI)** and generates a diagnostic report. This report provides insights into factors like soil health, crop condition, and resource efficiency.

5. **Update Agriculture Status**: Farmers can update their agricultural status by submitting new data, such as observed changes in crop health or environmental conditions. This data is stored in the **Data Base** and reviewed by both the **Agriculture Expert (AI)** and the **Agriculture Officer**, who can provide additional feedback or recommendations based on the updates.

6. **Ask Query**: Farmers can submit queries related to their agricultural concerns, such as pest infestations or irrigation issues. These queries are forwarded to the **Agriculture Officer [4]** for expert review and response.

7. **Answer Query**: The **Agriculture Officer** reviews the queries submitted by farmers and provides detailed answers through the system. The farmer then receives these answers, ensuring they have the necessary information to address their challenges.

This system of interconnected use cases and actors ensures that farmers receive comprehensive support, blending automated insights from AI with expert human advice to address a wide range of agricultural needs efficiently.

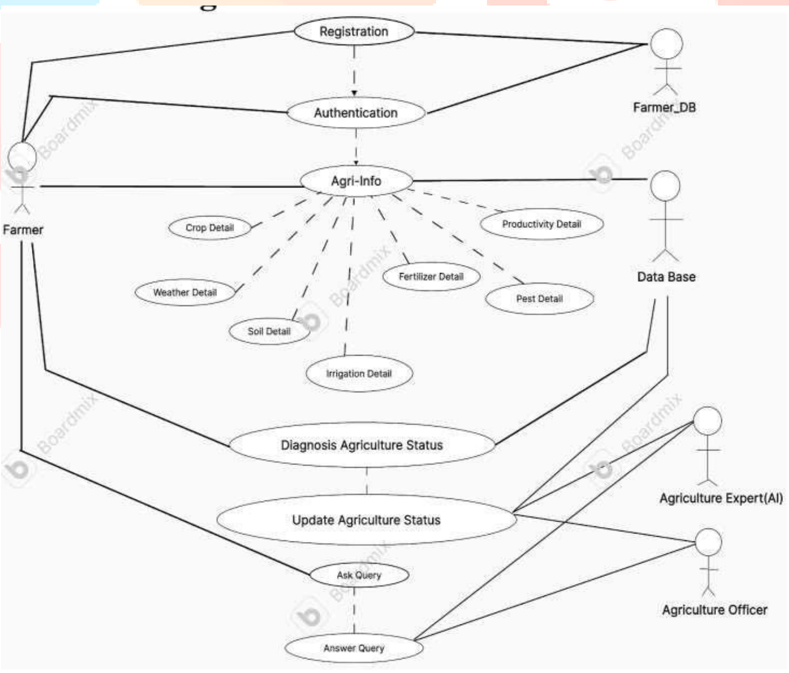

Figure 3: Use Case Diagram This diagram outlines the primary use cases of the system and their interactions with various actors, including farmers, databases, agriculture officers, and AI modules. It provides a visual representation of functionalities such as registration, authentication, crop recommendations, disease diagnosis, and query resolution, showcasing how users benefit from the platform.

IV. Implementation Details

The implementation of the proposed system involved several stages, starting from data preparation to the deployment of the platform. The dataset, primarily sourced from Kaggle, included a diverse range of agricultural data, such as NPK values, crop yields, environmental conditions, and plant disease images [1], [5]. To ensure the quality of the data, preprocessing techniques were applied, including handling missing values through mean imputation, normalization of attributes like NPK values, and splitting the dataset into training, validation, and testing subsets. For image-based data, preprocessing steps such as resizing to 250x250 pixels and data augmentation techniques (e.g., flipping, rotation) were applied to improve model robustness.

The system incorporated three main machine learning models to achieve its objectives. The crop prediction module used a Decision Tree Classifier trained on features like NPK values, temperature, and humidity to recommend suitable crops for specific conditions. For fertilizer recommendation, a regression model was implemented to predict the optimal quantities of fertilizers based on soil profiles. To diagnose plant diseases, a Convolutional Neural Network (CNN) was trained on a labeled dataset of healthy and diseased crop images, ensuring accurate disease detection and classification. The performance of these models was evaluated using metrics such as accuracy, precision, recall, and F1-score, ensuring reliability in real-world applications.

The frontend of the platform was developed using ReactJS to provide an intuitive and responsive user interface. This interface allowed farmers to input data, such as NPK values and crop images, and access predictions through a visually appealing dashboard. The backend, built with Flask, managed the server-side operations, including validating user inputs, interacting with machine learning models, and retrieving real-time weather data from external APIs. The backend communicated with the frontend through RESTful APIs, ensuring seamless data flow between components.

To enhance accessibility, an SMS notification service was integrated using Twilio, enabling farmers to receive daily weather updates and critical alerts directly on their mobile devices. This feature was particularly useful for farmers in areas with limited internet connectivity. The system was deployed on a cloud-based hosting platform, such as Render, ensuring scalability and availability. Deployment involved containerizing the application using Docker to streamline the setup process and ensure compatibility across environments.

V. Results and Evaluation

The machine learning models demonstrated strong performance across various metrics. The **Crop Prediction Model**, based on a Decision Tree Classifier, achieved an accuracy of 87%, indicating its reliability in recommending suitable crops based on NPK values, temperature, and humidity. Similarly, the **Fertilizer Recommendation Model** exhibited an R^2 score of 0.92, suggesting a high degree of precision in predicting optimal fertilizer quantities. The **Disease Detection Model**, implemented using a Convolutional Neural Network (CNN), achieved an accuracy of 91% on the test dataset, successfully classifying diseases from uploaded crop images. The evaluation metrics for all models, including precision, recall, and F1-score, highlighted their effectiveness in real-world applications.

VI. Conclusion

The proposed system successfully integrates machine learning, real-time weather data, and user-friendly interfaces to provide actionable insights for farmers, addressing critical challenges in precision agriculture. The platform's features, including crop prediction, fertilizer recommendation, and disease detection, empower farmers to optimize their resources, improve productivity, and minimize losses. Additionally, the SMS notification service enhances accessibility, making the system inclusive for users in remote or internet-constrained areas.

Despite its limitations, the system demonstrates a strong foundation for scalable and sustainable agricultural solutions. Future work will focus on expanding the dataset, improving model accuracy, and incorporating additional features such as pest control and yield prediction. By addressing these enhancements, the system has the potential to play a transformative role in modernizing agricultural practices globally.

REFERENCES

[1] F. H. Tseng et al., "A Decision Support Framework for National Crop Production Planning," *IEEE Access*, vol. 7, pp. 116965–116973, 2019.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] S. Godara et al., "AgriResponse: A Real-Time Agricultural Query-Response Generation System for Assisting Nationwide Farmers," *IEEE Access*, vol. 12, pp. 294–307, 2024.K. Elissa, "Title of paper if known," unpublished**.**

[3] A. Sharma et al., "Machine Learning Applications for Precision Agriculture: A Comprehensive Review," *IEEE Access*, vol. 9, pp. 4843–4852, 2021.

[4] G. Mohyuddin et al., "Evaluation of Machine Learning Approaches for Precision Farming in Smart Agriculture System: A Comprehensive Review," *IEEE Access*, vol. 12, pp. 60150–60159, 2024.

[5] "Applying Big Data for Intelligent Agriculture-Based Crop Selection Analysis," presented at the *Big Data and Agriculture Conference*, 2020.