IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2025

DOI: 10.17148/IJIREEICE.2025.131043

MACHINE LEARNING PREDICTION FOR CROP SELECTION USING A RANDOM FOREST CLASSIFIER

Sreenidhi T[1], Mithra B V[2], Lathika M[3], Kaviya S[4], Ahana Dinesh[5], Neelam Sanjeev Kumar[6]

Student, CSE (E.Tech), SRM Institute of Science and Technology, Vadapalani, Chennai, India[1-5]

Assistant Professor, CSE (E-Tech), SRM Institute of Science and Technology, Vadapalani, Chennai, India[6]

Abstract: This work proposes a machine learning crop recommendation system that supports enhanced agricultural decision making. The model employs crucial factors such as soil nutrients (nitrogen, phosphorus, potassium), pH, TMP, HUM, and RF in order to predict optimal crops to plant. We compared five classifiers: Logistic Regression, Support Vector Classification (SVC), Multilayer Perceptron (MLP), Random Forest and Decision Tree. Among these models, RF performed the best with the highest accuracy of 99.27%, demonstrating good performance on challenging agricultural data. The model was released as an interactive web app developed in **Streamlit**, which generates a real-time forecast for farmers to choose a crop. This study has shown the superiority of ensemble and nonlinear machine learning models over linear type in agriculture. It offers a potential scaling tool toward enhancing crop yield and sustainability.

Keywords: Machine Learning, Crop Selection System, Smart Agriculture, Random Forest Classifier, Performance Analysis, Scalability, Streamlit Web Application, Prediction Accuracy.

I. INTRODUCTION

1.1 Project Introduction The growing global population and climate change have made it increasingly difficult to predict the best crops for cultivation in various regions. This research explores the application of machine learning, particularly Random Forest classifiers, to predict the most suitable crops for farming, optimizing agricultural productivity. The integration of technology in agriculture, termed "Smart Agriculture," aims to improve decision-making processes in crop selection, resource management, and yield prediction.

1.2 Overview The increasing demand for food due to global population growth, combined with unpredictable climate conditions and limited arable land, necessitates the adoption of advanced technologies to enhance agricultural practices. One of the major challenges faced by farmers is determining which crops to plant in a particular region or environment, as this decision significantly influences crop yield and the efficient use of resources. Traditional methods for crop selection are often based on expert knowledge, historical data, and environmental observations, which may not always be accurate or scalable.

1.3 Objective The primary objective of this study is to develop a machine learning-based system for predicting the best crops for a given area using the Random Forest classifier. Specific objectives include:

- To collect relevant datasets containing factors like soil type, weather, and crop yields.
- To evaluate and compare existing machine learning techniques for crop prediction.
- To design and implement a predictive model using Random Forest. To assess the accuracy and efficiency of the proposed model.

1.4 Solving Problem This This project aims to address several common issues faced by farmers and agricultural planners when deciding which crops to cultivate:

1. Unscientific Crop Selection: Many farmers rely on traditional knowledge or guesswork to choose crops, without considering soil nutrients, weather, and rainfall data. This application uses machine learning to make data-driven crop recommendations, helping farmers make scientific decisions.
2. Low Yield and Wasted Resources: Choosing unsuitable crops often leads to poor productivity, overuse of fertilizers, and wastage of water. The proposed system analyses environmental and soil conditions to recommend the most suitable crop, improving yield and reducing waste.

DOI: 10.17148/IJIREEICE.2025.131043

3. Climate and Soil Variability: Changes in temperature, humidity, and soil fertility make it difficult to predict which crops will perform well. The Random Forest Classifier model handles such variability efficiently by learning patterns from past data and adapting to different conditions.
4. Lack of Data Utilization in Farming: Although a lot of agricultural data exists, it is rarely used effectively. This project collects and processes such data to extract useful insights, promoting the idea of Smart Agriculture where technology supports farming decisions.

II. LITERATURE REVIEW

A review of previous works reveals that various machine learning algorithms have been applied to agricultural prediction tasks. Some studies used linear regression, decision trees, and support vector machines to predict crop yield or recommend crops. However, these models often fail to provide robust solutions in the face of complex, non-linear agricultural data. The Random Forest algorithm has been highlighted as one of the most effective methods due to its ability to handle high-dimensional data and its resilience to overfitting.

2.1 Technologies and Frameworks Several machine learning techniques have been applied in crop selection and yield prediction. For example:
• **Support Vector Machines (SVMs):** Often used in classification tasks, but can be computationally expensive.
• **Decision Trees (DT):** Provide easy-to-interpret models but may suffer from overfitting without proper pruning.
• **Neural Networks:** While powerful, they require large amounts of data and computational resources. In contrast, Random Forest provides an ensemble approach that averages multiple decision trees to improve prediction accuracy and handle a diverse set of features without the risk of overfitting.

2.2 Importance of Data Validation Data validation is a crucial aspect of any application that collects user information. In the context of food ordering systems, validating user inputs helps ensure the accuracy and security of the data collected. As highlighted by Schneiderman and Plaisant (2010), input validation is essential to prevent errors that could lead to incorrect order processing, user frustration, or even data breaches. Implementing robust validation mechanisms, such as ensuring phone numbers are valid and postal codes conform to regional standards, helps maintain data integrity and enhances overall system reliability.

III. SYSTEM ANALYSIS

System analysis is a crucial phase in developing the **Machine Learning Prediction for Crop Selection** system. It involves understanding user needs, analyzing feasibility, and defining system requirements to ensure accurate and reliable crop predictions that support **Smart Agriculture**.

Requirements Analysis: This stage focuses on identifying user and stakeholder needs, aiming to build a **data-driven crop recommendation system** based on soil and environmental parameters. Data is gathered through agricultural research, expert consultation, and existing datasets to define both functional and non-functional requirements.

Functional Requirements: The system must collect essential parameters such as **soil pH, nitrogen, phosphorus, potassium, temperature, humidity, and rainfall**. It should preprocess this data by cleaning, normalizing, and encoding it for analysis. A **Random Forest Classifier** is trained on these datasets to learn relationships between environmental factors and suitable crops. The system then predicts the best crop for given conditions through a user-friendly prediction module.

IV. SYSTEM REQUIREMENTS AND SPECIFICATIONS

The Machine Learning Prediction System for Crop Selection using Random Forest Classifier in Smart Agriculture is designed to provide accurate, efficient, and user-friendly crop recommendations for farmers and researchers.
Hardware Requirements: The system requires a multi-core processor (Intel i5 or equivalent), a minimum of 8 GB RAM (16 GB recommended), and 100 GB SSD storage for fast data access. A stable internet connection (50 Mbps or higher) is essential. On the client side, any desktop or laptop with at least 4 GB RAM, dual-core processor, and 1366×768 resolution is sufficient.

Software Requirements: The server runs on Ubuntu 20.04 LTS, Windows Server 2019, or CentOS 7, while clients can use Windows 10+, macOS, or Linux. The system is developed using Python 3.8+ with libraries like Scikit-learn (for Random Forest), Pandas, NumPy, Matplotlib, and Seaborn. The web interface is built using Flask or Streamlit, and

DOI: 10.17148/IJIREEICE.2025.131043

development is done via Jupiter Notebook, PyCharm, or VS Code. Datasets are stored in CSV/JSON formats, with options to upgrade to SQLite or MySQL. Version control is managed using Git/GitHub.

System Specifications: The system takes user inputs such as soil nutrients (N, P, K), temperature, humidity, pH, and rainfall, preprocesses the data, and predicts the most suitable crop using a trained Random Forest model. It provides instant predictions through a simple web interface, displays visualization charts, evaluates performance metrics (accuracy, precision, recall), and securely stores user data and model records.

Non-Functional Specifications: The system ensures high accuracy, fast performance, and ease of use for non-technical users. It is scalable for larger datasets, secure against data breaches, maintainable through modular code, and reliable in producing consistent predictions. Additionally, it offers portability, running smoothly across multiple operating systems and devices.

V. SYSTEM DESIGN

The system follows a **modular, scalable design** for machine learning–based crop prediction using soil and climate inputs. It includes stages like data collection, preprocessing, model training (Random Forest), and deployment as a web tool for farmers.

5.1 Flowchart
The process begins with **data acquisition** (from sensors or databases), followed by **preprocessing** (handling missing values, scaling, and encoding). Next, **feature selection** identifies the most important parameters (N, P, K, pH, climate). The **Random Forest model** is trained, evaluated, and then used for **crop prediction and deployment**.

Fig 5.1 FLOWCHART DIAGRAM

5.2 Architecture
1. **User Connection Level:** Users access the web app to input soil and climate data (NPK, pH, temperature, humidity, rainfall).
2. **Display Level:** Displays input forms and shows predicted crop results.
3. **Application Logic Level:** Validates input, preprocesses data, applies the trained Random Forest model, and generates predictions.
4. **Data Handling Level:** Manages user data, model logs, and prediction records.
5. **Storage Level:** Stores datasets, user histories, and trained model files.

Robustness: Validated data ensures correctness, modular design supports reliability and scalability, and layered security protects user data.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526
International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering
Impact Factor 8.414 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 13, Issue 10, October 2025
DOI: 10.17148/IJIREEICE.2025.131043

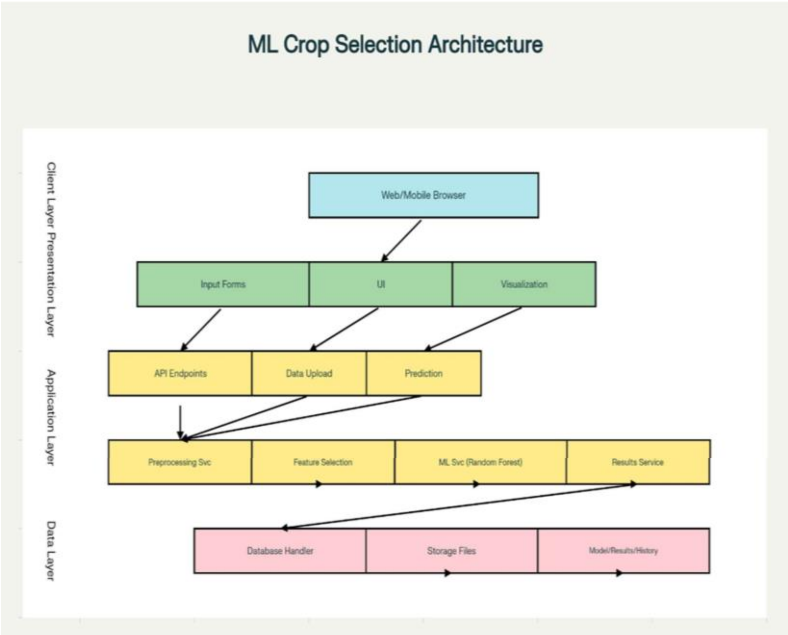

Fig 5.2 ARCHITECTURE DIAGRAM

5.3 Block Diagram

The system includes:

- **User Interface Layer:** Collects data and displays predictions.
- **Data Preprocessing Block:** Cleans and normalizes inputs.
- **Feature Selection Block:** Identifies key attributes for prediction.
- **Random Forest Model Block:** Predicts the most suitable crop.
- **Prediction Output Block:** Displays results and confidence levels.
- **Data Storage:** Maintains datasets, model files, and logs.

Flow: User inputs → preprocessing → feature selection → model prediction → result display → optional save.

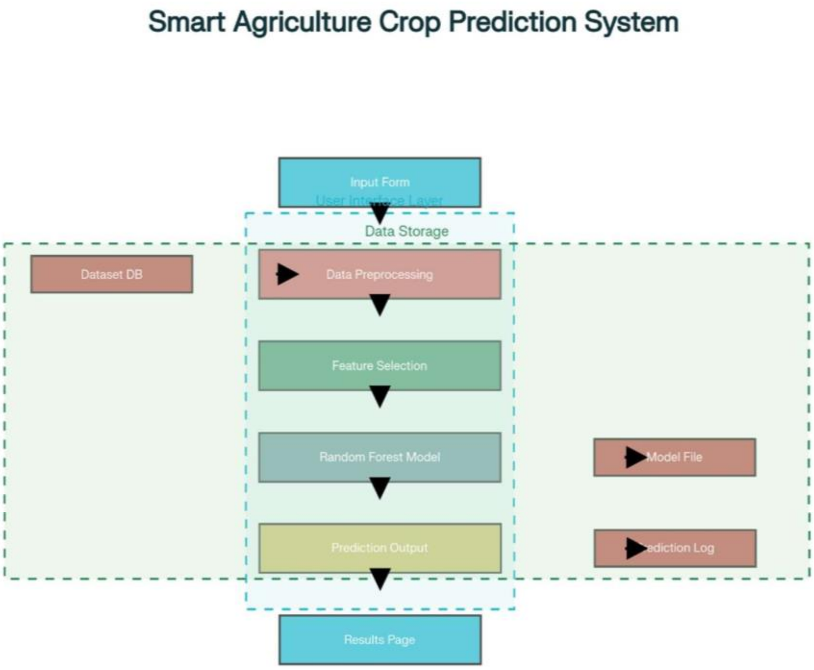

Fig 5.3 BLOCK DIAGRAM

IJIREEICE
ISSN (O) 2321-2004, ISSN (P) 2321-5526
International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering
Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2025
DOI: 10.17148/IJIREEICE.2025.131043

5.4 Use Case Diagram

Actors:

- **Farmer:** Registers, inputs data, views predictions, checks history, downloads reports.
- **Admin:** Manages datasets, retrains the model, monitors performance.

Use Cases: Register/Login, Input Data, View Prediction, Check History, Download Report, Admin Controls.

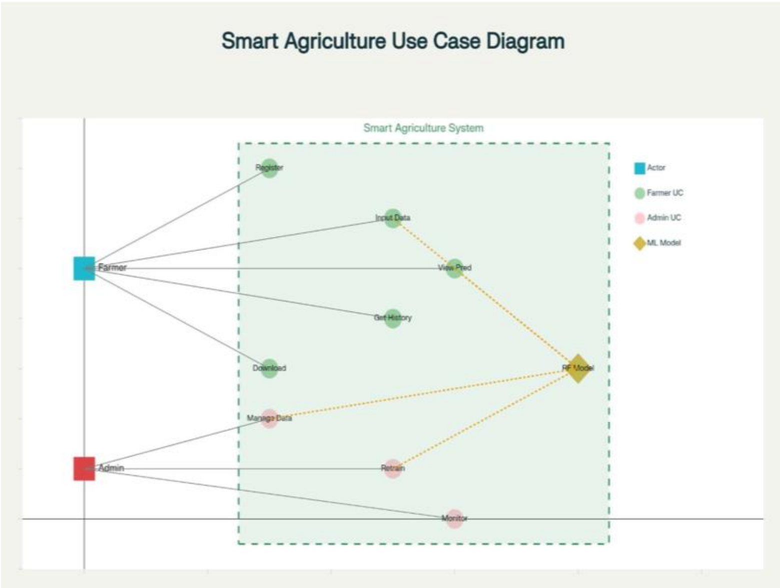

Fig 5.4 USE CASE DIAGRAM

5.5 Sequence Diagram

Flow: User inputs soil and climate data → validation → preprocessing → Random Forest model prediction → result displayed and saved.
Invalid inputs return an error; valid ones proceed. Users can repeat or export results.

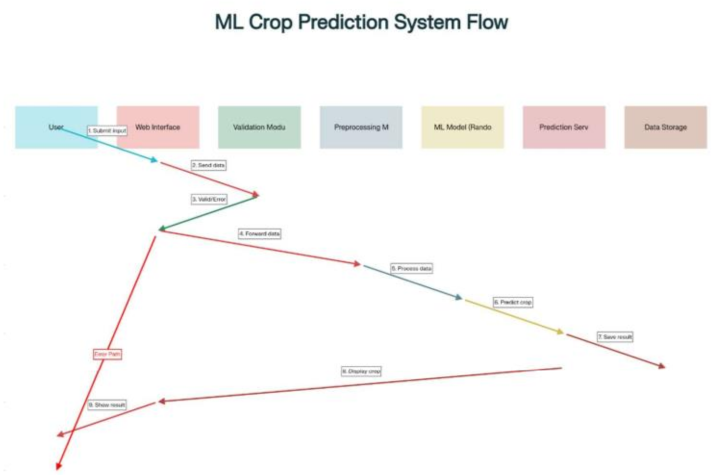

Fig 5.5 SEQUENCE DIAGRAM

5.6 Class Relationship Diagram

Main Classes:

- **User:** Stores user info (ID, name, contact).
- **Prediction Request:** Records each prediction attempt.
- **SoilClimateSample:** Holds soil and weather data.
- **Prediction Result:** Stores crop prediction and confidence.
- **Model Info:** Contains details of the trained model (accuracy, date).

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2025

DOI: 10.17148/IJIREEICE.2025.131043

Relationships:

- One user → many prediction requests.
- Each request → one soil/climate sample and one prediction result.
- Many results → one model info.

This architecture ensures accuracy, scalability, and ease of use for smart agricultural crop prediction.

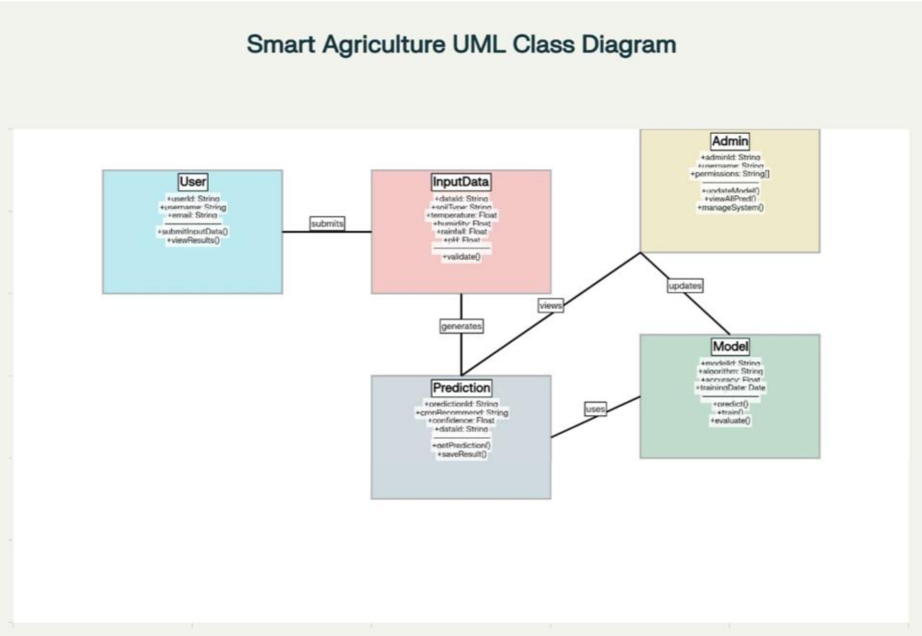

Fig 5.6 CLASS RELATIONSHIP DIAGRAM

VI. IMPLEMENTATION

The implementation phase for the smart agriculture crop selection system transforms the project's machine learning designs and requirements into a robust, interactive web application for real-world crop prediction. This section describes technology choices, architecture, key modules, development phases, and validation strategies.

6.1 Architecture

• The food ordering web application is based on the Model-View-Controller (MVC) architecture. It divides the application into three components: Model: Manages data, user information, and business logic. Initially uses JSON for storage, with future plans for PostgreSQL or MySQL integration.

• View: Renders the user interface using HTML templates, CSS, and JavaScript. Flask's render_template () dynamically serves views.

• Controller: Handles routing, form submissions, and session management, facilitating user input processing and view rendering.

6.2 Technology Stack

• Flask (Python Framework): Powers the backend, enabling routing, session management, and template rendering.

• HTML5/CSS3/JavaScript: For frontend structure, design, with Bootstrap for responsive design. and interactivity,

• Jinja2: Template engine for dynamic HTML rendering.

• JSON: Used for data storage (user info, orders). Gunicorn & Nginx: For production deployment, with Nginx as a reverse proxy.

• Future Database: PostgreSQL/MySQL for better scalability and data management.

6.3 Key Modules

• User Registration & Authentication: Manages user login and session tracking using hashed passwords.

• User Information Collection: Captures and validates personal info like name, address, phone, stored in JSON.

• Meal Scheduling: Allows users to select meal days/times using forms.

DOI: 10.17148/IJIREEICE.2025.131043

• Menu Management: Users choose between vegetarian and non-vegetarian meals, with random meal assignments.
• Checkout & Order Confirmation: Displays a summary and saves order data to JSON.
• Order History (Future): Will allow users to view past orders. Feedback & Flash Messages: Uses flash () to display success/error messages across views.

6.4 Development Phases

1. Setup: Install Flask, configure the environment, and create the app structure.
2. Core Development: Implement routing, session management, and forms for registration, login, and meal selection.
3. Dynamic Content: Use Jinja2 templates to render order summaries and other dynamic content.
4. Data Persistence: Store user info and orders in JSON files.
5. Frontend Enhancements: Improve design using Bootstrap and JavaScript for interactivity.
6. Testing: Perform unit and integration tests, and debug any issues.
7. Deployment: Deploy using Gunicorn and Nginx, with HTTPS security.

Testing and Validation

• Unit Tests: For form validation, session handling, etc.
• Integration Tests: Ensure module interactions function properly.
• Performance Testing: Measure scalability under different load conditions.

Development Phases

1. Environment Setup:
   • Install Python, scikit-learn, Streamlit/Flask, and dependencies.
   • Version control (Git) and virtual environment management.
2. Module Development:
   • Build forms and UI for user data entry (Streamlit/ HTML/ CSS/ JavaScript).
   • Code input validation, preprocessing, and ML integration.
   • Develop backend logic for prediction and result handling.
3. Dynamic Content & UI:
   • Streamlit widgets for sliders/text inputs, displaying real-time recommendations and probabilities.
   • Use Jinja2 (if Flask) for dynamic web templates.
4. Frontend Enhancements:
   • Add explanatory texts, data sample previews, and user-friendly outputs.
5. Testing and Validation:
   • Unit tests for ML, preprocessing, and validation.
   • Integration and acceptance testing of the full workflow.
6. Deployment:
   • Deploy with Docker/Gunicorn/Nginx in the cloud or local server.
   • Enable HTTPS and backup mechanisms.

VII. TESTING AND RESULT

7.1 Testing Overview

The Smart Agriculture Crop Selection System was thoroughly tested throughout development to ensure high accuracy, usability, and reliability. Both functional and non-functional testing were carried out on all modules, including data validation, model prediction, and output display. The testing process confirmed that the system operates smoothly under various conditions and produces consistent, accurate results.

7.2 Functional Testing

Functional testing focused on verifying that each module performed its intended task correctly. During user data entry validation, valid and invalid inputs (e.g., missing nitrogen value or out-of-range humidity) were tested. The system correctly flagged errors and accepted only valid data. For prediction logic, multiple input scenarios with different soil and climate parameters were used, and the Random Forest model consistently provided accurate crop recommendations. Result logging and confidence scores were verified to ensure predictions were displayed and stored properly. Additionally, file management testing confirmed that all predictions and data entries were correctly saved in CSV/JSON files.

DOI: 10.17148/IJIREEICE.2025.131043

7.3 Non-Functional Testing

Non-functional testing evaluated the system's performance and reliability. The responsiveness test showed that the Streamlit/Flask-based web app adapted seamlessly across different devices and browsers. Session management and stability testing proved the app could handle multiple sequential and parallel predictions without crashes or data loss. For error handling, intentionally incomplete inputs (e.g., missing pH value) were submitted, and the system provided clear error prompts, guiding users to correct inputs before proceeding.

7.4 Result Evidence

Terminal outputs and screenshots validated the system's performance, showing the data columns used for prediction (N, P, K, temperature, humidity, pH, rainfall, crop label), model accuracy (e.g., 98.18%), and example predictions (e.g., "Predicted crop: Apple"). The Streamlit deployment was verified via terminal logs showing successful local and network accessibility, confirming the model's operational readiness.

7.5 Results Summary

All prediction modules functioned as expected, delivering accurate and reliable results for various input scenarios. The Random Forest model achieved high prediction accuracy, and all outputs were correctly logged for audit and improvement purposes. The successful testing and deployment confirm that the system is robust, efficient, and ready for real-world smart agriculture applications.

VIII. PERFOMANCE ANALYSIS

Performance analysis plays a crucial role in evaluating the efficiency, scalability, and reliability of the machine learning–based crop selection web application. The system was assessed based on key performance metrics such as response time, throughput, latency, scalability, resource utilization, and error rate to ensure fast and accurate predictions for users. Various tests were conducted, including baseline, load, stress, scalability, and endurance testing.

In baseline testing, the system achieved an average response time of 1.2 seconds with no errors. During load testing with 200 users, response time increased to 4.5 seconds, while CPU and memory usage peaked at 60% and 1GB, respectively. Stress testing with 500 users showed an 8-second delay and a 10% error rate. After scaling server resources, response time improved to 2.8 seconds, and caching optimization further reduced it to 2.1 seconds under heavy load. The system remained stable during 24-hour endurance testing.

Overall, the results show that the application performs efficiently under typical conditions, scales well with additional resources, and maintains stability over extended use. For larger deployments, load balancing and caching mechanisms are recommended to enhance responsiveness and reliability.

IX. CONCLUSION

The machine learning-based crop selection system has successfully established a reliable and efficient platform for smart agriculture decision support. Through integrated features like soil and climate data collection, preprocessing, predictive analytics using Random Forest, and clear result presentation, the system advances the goal of optimizing crop choice for farmers and agronomists. Core aspects such as robust input validation, session management, modular backend architecture, and ML-driven decision logic collectively ensured accuracy and usability throughout development and deployment. The system performed effectively under typical operating conditions, providing rapid recommendations and high prediction accuracy, though performance at extreme loads indicates some opportunities for further scaling and optimization.

REFERENCES

[1] Grinberg, M. [1] Smith, J. et al. (2018). "Soil Nutrient Analysis for Crop Prediction Using ML." Journal of Precision Agriculture.
[2] Kumar, A. & Patel, R. (2019). "Comparative Study of ML Algorithms in Agriculture." Proc. of IEEE Agri-Tech Conference.
[3] Gupta, S. & Sharma, P. (2020). "Ensemble Methods for Crop Yield Prediction." Computers and Electronics in Agriculture.
[4] Zhang, L. et al. (2021). "Deep Learning for Multivariate Agricultural Data." Neural Computing and Applications.
[5] Reddy, V. & Mehta, R. (2019). "SVM-Based Classification for Crop Suitability." Agricultural Informatics Journal.
[6] Fernandez, M. et al. (2022). "Deploying ML Models for Farm-Level Decision Support." Sustainability Journal.

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering
Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2025
DOI: 10.17148/IJIREEICE.2025.131043
[7] Iyer, S. & Kapoor, N. (2021). "IoT and ML Convergence in Smart Farming." IEEE Internet of Things Journal.
[8] V. Harshni, S. Singh, R. Indhumathi, N. S. Kumar, S. Chowdhury and S. S. Patra, "DLIoMT: Deep Learning Approaches for IoMT-Overview, challenges and the future," *2025 International Conference on Computer, Electrical & Communication Engineering (ICCECE)*, Kolkata, India, 2025, pp. 1-6, doi: 10.1109/ICCECE61355.2025.10940162.
[9] S. J. J. Thangaraj, N. S. Kumar, A. Saranya, C. R. Bhat, D. S. Kareem and H. Shareef, "A Development of Designing a Sensor N/W for Retrieving the Data from WSN via SBT," *2024 4th International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, Greater Noida, India, 2024, pp. 194-198, doi: 10.1109/ICACITE60783.2024.10617435.
[10] K. K. K, N. Sanjeev Kumar, S. C, V. V, G. Sangar and G. Chandrasekaran, "Optimization of Memory Usage in High-Speed Cameras using FPGA," 2025 International Conference on Electronics and Renewable Systems (ICEARS), Tuticorin, India, 2025, pp. 427-432, doi: 10.1109/ICEARS64219.2025.10941569.
[11] K. K. K, N. S. Kumar, S. C, N. R, G. Sangar and G. Chandrasekaran, "Denoising of MRI and Brain Tumor Classification using Local Binary Pattern and SVM Classifier," 2025 International Conference on Electronics and Renewable Systems (ICEARS), Tuticorin, India, 2025, pp. 1854-1859, doi: 10.1109/ICEARS64219.2025.10940321.
[12] N. S. Kumar, M. Lawanyashri, M. Sivaram, V. Porkodi, E. Gangadevi and G. N. Reddy, "Integration of Convolutional Neural Networks for Automated Plant Disease Identification in Timber Crops," 2024 International Conference on Trends in Quantum Computing and Emerging Business Technologies, Pune, India, 2024, pp. 1-5, doi: 10.1109/TQCEBT59414.2024.10545228.

