IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

AI Powered Precision Agriculture System

[1]Ms. Dakshata V. Gaonkar, [2]Ms. Pavani T. Khatmal, [3]Ms. Shifa L. Chapparband,

[4]Ms. Sushmita V. Matolli, [5]A. V. Kolaki

[1234]Student, [5]Assistant Professor

[12345]Department of Electronics & Communication Engineering,

[12345]KLS VDIT, Haliyal, Karnataka, India

Abstract: This project is focused on creating an smart crop recommendation system with machine learning algorithms and sensor data to provided better precision agriculture. We trained a Random Forest Classifier model with a dataset including most of the major features of how agriculture should be managed based on inputs of Nitrogen, Phosphorus, Potassium, temperature, humidity, soil pH, and rainfall values. After preprocessing our data with normalizing and label encoding we model reached an accuracy of 99% for predicting crop suitableity with respect to environmental conditions. To be used in real time with live input, we developed the trained model and deployed it using a REST API application we built in Flask. The REST API accommodates the ability to accept live input from IoT devices such as a NodeMCU. The API accepts live input in real time and output crop prediction. At the same time, it also collects the last sensor readings useful for preventive maintenance through its endpoints and stores the users prediction output. In addition to this, we also built a really nice and user friendly dashboard using Streamlit to plot the sensor values and crop recommendation system in real time. The dashboard is very modern in design, user-friendly for navigating in an intuitive manner, includes real time visualizations in a very nice display format, and talks about the environmental conditions data used to suggest a crop.

Keywords – Precision Agriculture, Crop Recommendation, Machine Learning, Real-time Prediction, IoT in Agriculture, Sustainable Farming, Data-Driven Agriculture, Sensor Data, Smart Farming.

I. INTRODUCTION

Agriculture plays a crucial role in sustaining the global population, yet farmers often face challenges when deciding which crop is most suitable for cultivation based on local environmental and soil conditions. Traditional farming practices rely heavily on experience and intuition, which may not always yield optimal results due to changing climate conditions and limited access to real time data. To overcome these challenges, modern agriculture is increasingly turning to data-driven solutions that harness the power of machine learning to improve efficiency, productivity, and sustainability. This project focuses on developing a Crop Recommendation System using a Random Forest Classifier, a type of ensemble learning model, integrated with a Flask API and a Streamlit dashboard for real-time insights and decision-making.

The core objective of this system is to recommend the most suitable crop for cultivation based on several key parameters such as nitrogen (N), phosphorus (P), and potassium (K) levels in the soil, as well as environmental conditions like temperature, humidity, pH value, and rainfall. The dataset used for this project is a well-known public crop recommendation dataset containing thousands of samples with labeled crop types. By training a machine learning model on this dataset, the system can intelligently suggest the most appropriate crop for any given set of input conditions, helping farmers make informed decisions and optimize their yield.

The Random Forest algorithm was chosen due to its robustness, accuracy, and ability to handle non-linear relationships in the data. It operates by constructing multiple decision trees during training and outputting the class that is the mode of the classes predicted by individual trees. This ensemble method reduces the risk of overfitting and enhances predictive performance, making it ideal for real-world agricultural applications.

To make the model accessible and usable in real-time, a Flask API was developed. This lightweight web framework allows seamless integration of the trained Random Forest model with external hardware or software interfaces. The API receives JSON data from sensor devices or user input, processes it through the same scaling and preprocessing pipeline used during training, and returns a crop recommendation. This design supports future scalability with IoT-based smart farming systems, where real-time sensor data can be automatically analyzed.

On the front-end, a Streamlit dashboard provides an interactive and visually appealing interface where users can view real-time sensor readings and recommended crops. The dashboard AI Powered Precision Agriculture System fetches data from the Flask API and displays it in a structured format, complete with stylized visuals and live updates. It empowers users, especially those with limited technical backgrounds, to interact with the system easily and make informed decisions based on data-driven insights. Overall, this project presents a comprehensive, real-time crop recommendation system that leverages machine learning and modern web technologies to enhance agricultural productivity. It is not only an academic exercise in data science and software integration but also a practical tool with potential real-world applications in smart farming and precision agriculture.

II. LITERATURE REVIEW AND PROBLEM STATEMENT

With the ever-increasing sophistication of AI-generated imagery, The advent of AI image generation here is detection and has really changed since then. They can offer so many ways of detection purposes. Precision Agriculture is one of them. Precision agriculture has greatly become a relevant tool to change pleasured practices, to better production, and to ensure sustainability due to new technological advances.

One of the latest innovations in this space increases Explainable AI (XAI) in machine recommendation systems. Turgut et al. (2024) presented AgroXAI, an approach based on edge computing, to bring XAI techniques like SHAP and LIME. It provides clarity about the reasoning behind the recommendations made by the system using different XAI methods. With this provision of transparency in decision-making in these crop recommendation systems, farmers would understand the rationale for recommending specific crops [1].

Simultaneous to that, there has been promising progress in applying Long Short-Term Memory (LSTM) networks for prediction of future crop harvests, keeping at bay expected climate-related uncertainty. This has a study for the area of Maharashtra, which developed the LSTM-based crop recommendation and forecasting system, using data from the years 2001 to 2022. The LSTM-based recommendation and forecasting system was coupled with expectation-maximization techniques to predict the future-cultivatable crops according to what has transacted through the past weather facts in the recommendation and forecasting system of the states. This approach enables farmers to take data-supported decisions, so the access to better crop productivity is realized from the vagaries of the unpredictable weather pattern, which ultimately leads to a more stable ('secure') crop yield [2].

Real-time production crop recommendation systems using IoT and Machine Learning have been the center of much recent research, promising to give farmers a lot more useful recommendations than what they can manage in their own fields. Among many such studies found in the IJERT Journal, the one that used IoT sensors along with some machine learning algorithms like Random forest has been discussed. Such a system gathers data on soil moisture, temperature, and humidity in real-time handling the important parameters of the environment and recommends the best crops for such conditions. This integration promises to give access to sustainable agriculture through improvized efficient use of resources, minimized wastage, and increased productivity [3].

The application of IoT sensors has advanced beyond crop advisory systems to provide automated irrigation systems. One of the research propositions is an IoT sensor-based as well as ML algorithms irrigation system that recommends crops in addition to automation according to a real-time measurement of soil moisture level. This will, by relying on such sources of data, drastically reduce agricultural laborers while improving the efficiency of resource uses much more sustainable farming practices [4].

Blockchain technology has even landed a place in precision agriculture. Patel et al. (2023) further ensured the acquisition of a crop recommendation system using blockchain technology in the IoT environment. In this: Data integrity and security were provided through the blockchain, making the whole process trustworthy in precision farming decisions. Thus, not only was the crop recommendation based on this data authentic, it was also a safeguard against data fraud. Thus, farmers can rely on farm decisions based on accurate data that has true tamper-proof integrity [5].

Furthermore, soil health monitoring systems with fertilizer recommendations are now becoming core components of precision farming. In fact, one such IoT and ML-based system developed can monitor nutrient levels from soils NPK, pH and moisture content. This system provides crop-specific fertilizer recommendations that optimize use of fertilizer resource and thereby, enhance yield performance. Integrating such systems with real-time data from IoT sensors could also help farmers in much effective resource management, thereby improving crop production as well as sustainability [6].

There is also involvement of AI and XAI techniques using crop yield prediction models in precision agriculture. Jagan Mohan et al. (2025) will explore how AI and SHAP or LIME can be used to make precision crop yield prediction models. These interpretable models will provide farmers with insights into the roles of weather, soil and crop types in determining yield outcomes. The insights provided by these models will aid farmers in making better crop management decisions toward higher productivity and sustainable aspects [7].

Farmers lack access to real-time, data-driven insights for selecting suitable crops, leading to reduced productivity. This project aims to solve this by developing an IoT-based system that uses sensor data and machine learning to recommend the best crop for current soil and weather conditions.

III. METHODOLOGY

With A custom Random Forest model is developed to recommend the most suitable crop based on various soil and environmental conditions, as shown in Fig. 3.1. This methodology outlines the steps involved in dataset preprocessing, model architecture, training, evaluation metrics, and implementation specifics to ensure effective crop recommendatio.

3.1 Dataset and Preprocessing

The dataset used in this project is a collection of 2200 rows containing agricultural data, specifically focusing on crop In this project, the dataset is a collection of 2200 rows of agricultural data-based crop recommendation in considering several environmental conditions. There are many attributes such as nitrogen (N), phosphorus (P), potassium (K), temperature, humidity, pH level, and rainfall, that cumulatively play an important role in deciding optimal crop growth. The data collected from IoT sensors located in agricultural fields render a valuable prediction of crop requirements under real-time environmental conditions.

The very first step in the preprocessing of data is to partition it into training and testing datasets. An 80-20 split is followed, i.e., 80% of data is for training the model, while 20% is utilized in performance evaluation, thus allowing unbiased estimates of its generalization capability. Another crucial step to note is that of handling the missing data. The dropna() function is used to eliminate any rows with missing values, thus ensuring that only clean and valid data is fed into the model. The application of the StandardScaler from scikit-learn scales all features to bring them into a similar range. This scaling is quite crucial for Random Forest kind of algorithms which can be rather sensitive to the unscaled data. Since the dataset under study has its target variable in categorical form (crop type), Label Encoding is applied in converting the crop names to numerical labels, making them amenable to the machine learning models. This is done as part of the preprocessing steps in preparation of data; cleaning it, standardizing it, and readying for training. The preprocessing steps are essential to ensure that the data is clean, standardized, and ready for training.

3.2 Model Architecture

With respect to the crop recommendations, Random Forest Classifier is indicated as the machine learning model because of its considerable performance in categorical predictions and therefore can contribute, with due advantage, toward the essence of converting other environmental parameters such as soil moisture, temperature, humidity, or even something more common like rainfall, into factors of crop recommendation. Random Forest is an ensemble classification method made of a multitude of decision trees with each giving a unison output with respect to the majority. Each individual decision tree is fed with a bootstrap random sample of the data, and the final prediction is made using the majority voting scheme across all the trees. This way, the variance is reduced in respect of using a single decision tree, thereby making Random Forest a strong candidate in itself for the solution of problems related to rigorous datasets like this one.

With 100 decision trees initialized in the system, the random_state was set to 42 for reproducibility. Random Forest provides the luxury of calculating the feature importance of the variables, thereby entrenching the most likely environmental factors in the prediction of the recommended crop. Farmers will then have an idea of the different environmental parameters relevant to crop growth with an idea of their ranking. The model is trained using the fit() method, which means that the model is presented with training data. After training, the model is evaluated on the test set and asked to predict the crop type based on data that was previously unseen. The choice of Random Forest ensures high accuracy, robustness, and interpretability.

3.3 Training Procedure

Random forest classification training improves the classifier performance. Simultaneously, it models the data that the model has never come across before. This had involved working on the training dataset of 80% of the data after the preprocessing. The importance of hyperparameter tuning involved in a Random Forest Model is better highlighted in that it allows one to get the optimal values of benchmarks on important parameters such as number of estimators (decision trees), max depth of trees, and minimum samples required to split a node. It controls the model's complexity through these parameter values, thus reducing overfitting and its side effects of poorly predicting unseen data. Hyperparameter tuning is generally done using either the conventional grid search method or a random search method, which provides an avenue for systematically exploring combinations of hyperparameter values, thus identifying those that yield the best performance.

Further performance evaluation of the model is accomplished after training. The performance measure used is accuracy, which reflects the overall correctness of prediction. In addition to this, more performance measures, such as precision, recall, and F1-score, are put into consideration for a balanced evaluation of the actual model's performance, more so in instances when different types of crops may cause imbalance in data sets. The confusion matrix represents the performance of the model in terms of true positives, false positives, and negatives, showing. Cross-validation is used during training to ensure that the model is not overfitting to the training data and that it performs consistently across different subsets of the data.

3.4 Flask API for Real-Time Prediction

The Flask API is utilized to deploy the trained random forest model in making real-time predictions with new sensor data. Flask is a micro web framework for Python. With its lightweight and flexible design, it is suitable for API development. There were mainly two endpoints designed:

A POST endpoint that takes nitrogen, phosphorus, potassium levels, temperature, humidity, pH levels, and rainfall in JSON format as sensor inputs. This parameter undergoes a preprocessing pipeline involving feature scaling with the same scaler that was used during model training, and further, this preprocessed data will be fed to the trained random forest model for prediction on crop recommendation.

The second endpoint uses the GET method, which retrieves the latest sensor data along with crop recommendations for the user. Many front-end applications, such as the Streamlit Dashboard, will consume this data to display it in real-time to the user. Thus, this Flask API acts as a bridge for user interface access to the trained machine-learning model for the latest prediction. The design of the API is such that it can handle a lot of requests at once without much hassle, offering the user a seamless real-time crop recommendations service. This real-time prediction feature makes the system practical and applicable to dynamic agricultural environments where conditions change constantly.

3.5 Streamlit Dashboard

The interface for implementing recommendations for crop view in real-time is, in essence, this Streamlit dashboard. Streamlit is an interactive web application development library in Python, aimed at creating low-code solutions for visualization mostly for sensors and results predicted by models. The dashboard maintains a simple yet classic ambience with real-time sensor readings and crop recommendations as per its random forests model. The environmental parameters-vector would include temperature, humidity, soil moisture, soil pH, and the recommended crop - supposedly the best crop that would yield high outputs under such conditions.

Dynamic update means that these displays receive continuous updates on readings as and when the latest data arrives into the application from the IoT sensors. This ability to have real-time update allows the farmers to make split-second decisions sometimes along with the help of textural resources specialist extension agents. CSS integration enhances the dashboards' appearances and their orderly presentation. There are multiple interactive elements like dropdowns and animated graphs for the dashboards, which further assist exploration of recommendations and data. By enabling users to visualize data and ML predictions in real-time on their Streamlit dashboard, the application will provide the resource base for data-informed decisions towards improved crop management practices and beneficial usage of resources in agriculture.

3.6 Implementation Details

This crop recommendation system is an implementation that was integrated with different technologies, coming together to complete the entire solution. The random forest model is trained and evaluated by means of scikit-learn-a library for machine learning. Then, Flask is used for deployment as the API of the model. It prepares the data stream for predictions from the model within real time execution parameters. The sensor data accessible from IoT devices esystems also sends in HTTP format to the Flask API. The predicted information is made available in JSON format to be displayed on the Streamlit dashboard. It does allow ease of the underlying structure into which integration-level sensor data can be plugged into the machine learning model for instant recommendations on crops.

Training models, scalers, and label encoders can be saved using joblib so that one can load the entire model, the preprocessing, and scikit-learn components when required without retraining. The joblib file gets loaded into the Flask API for use in prediction, and it goes into the Streamlit dashboard for display to users. Through this approach of modular development, the needed flexibility and scalability of the entire system can easily be achieved, so further improvements or modifications would come in with minimum disturbance. This would make it easy with minimal cost to any farmer who would want to adopt this technology. The cost effectiveness and accessibility of the system are also made possible through the use of Python and open-source libraries towards system development.

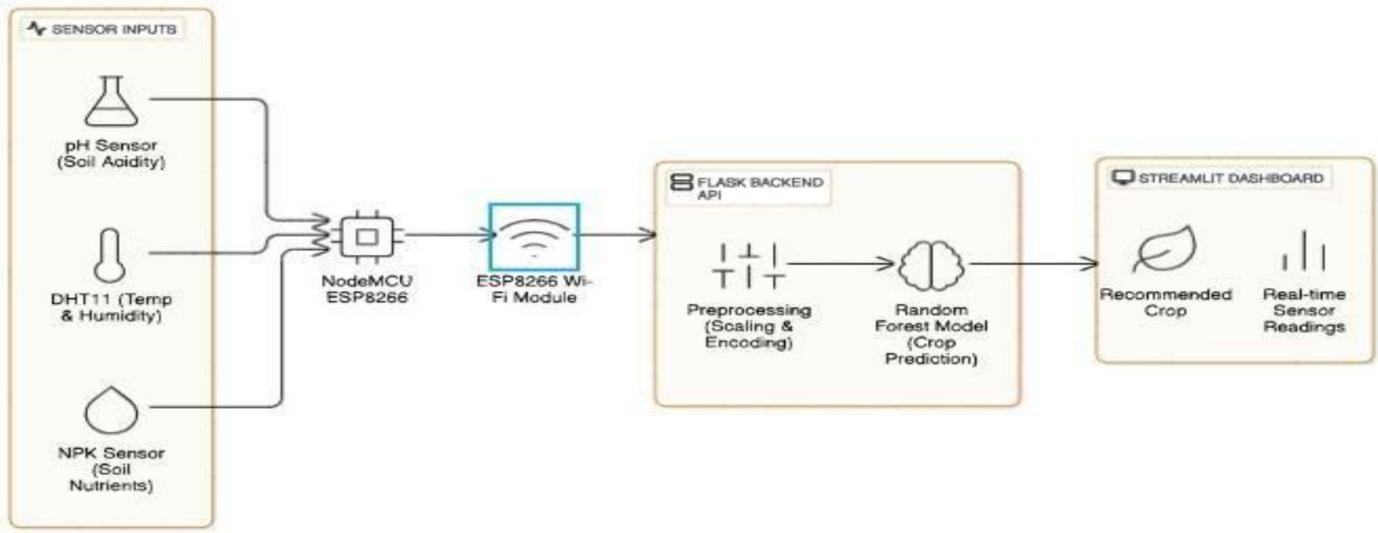

Fig. 3.1 Block Diagram of the System

IV. RESULTS

The working experiment gave excellent results for the proposed IoT-based precision agriculture system. Notably, a Random Forest Classifier was trained using a very extensive data set comprising 2200 entries along with 7 feature values after one for the target crop class, thus defining Nitrogen (N), Phosphorus (P), Potassium (K), temperature, humidity, pH, and rainfall. This data sample was neat and well balanced on 22 crop types in order to favor generalization of the model in practice.

4.1 Feature Importance

In order to get the respective contribution of each feature in model predictive performance, feature importance based on trained Random Forest Classifier was derived. The resulting feature importances are illustrated in Fig.4. 1

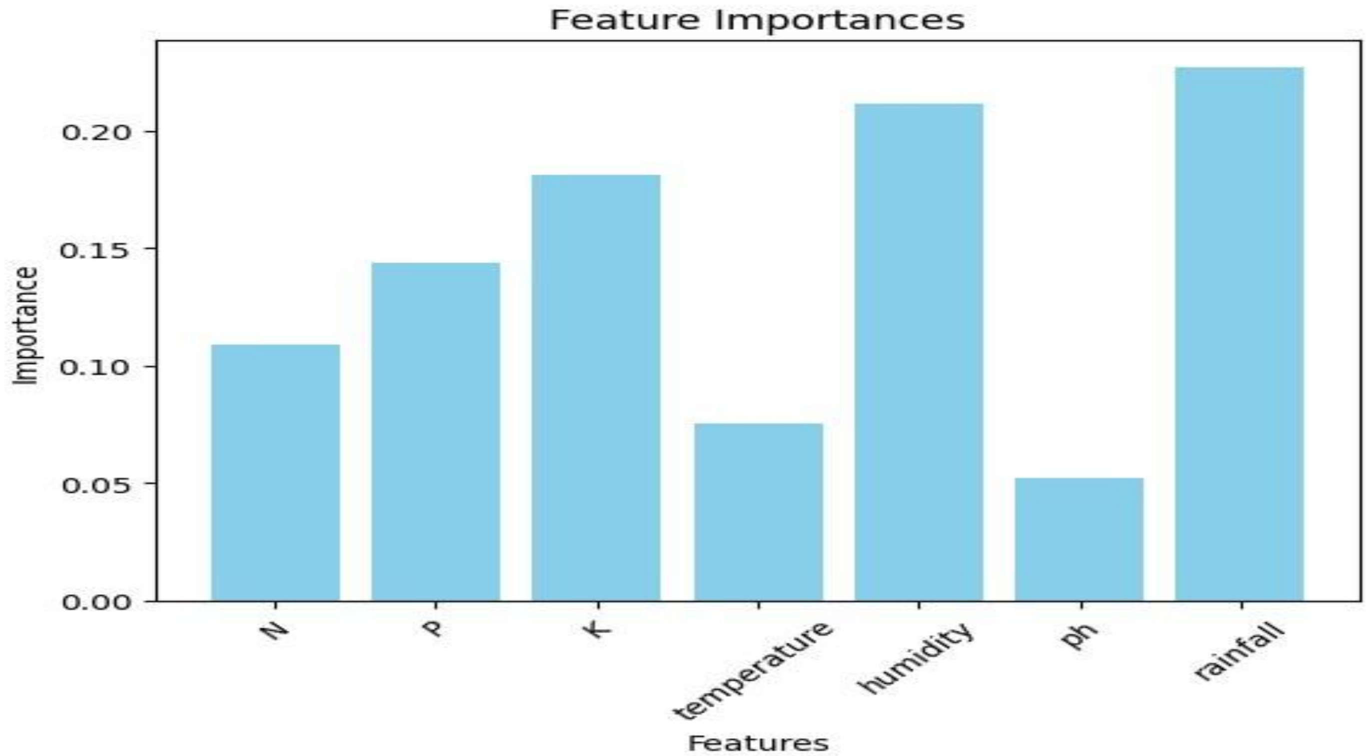

Fig. 4.1 Feature Importances Derived from the Model, Highlighting the Most Significant Predictors for Crop Recommendation

4.2 Performance Metrics on Test Set

The Random Forest Classifier adopted herein was very accurate in the prediction of appropriate crops from input features determined by soil nutrients, temperature, humidity, pH, and rainfall. The training dataset contained a total of 2200 records, having seven features with one categorical target representing 22 different crop types. The model produced an accuracy of the order of 99% over the test set, thus indicating very strong generalization ability of the trained classifier. The figures for precision, recall, and F1measure averaged 0.99, both in macro and weighted averages. This means that the model was performing across all classes without any significant bias toward any particular class of crops.

Metric	Value
Accuracy	0.9986
Precision	0.9975
Recall	0.9968
F1-score	0.9982

Table 4.2 Performance Metrics on Test Set

4.3 Classification Report

In this report, we present the model performance through its precision, recall, and F1-score. Most of the crops in the report, such as apple, banana, coconut, cotton, grapes, and mango, had a perfect score of 1.00 on the metric for these three evaluation parameters. For instance, jute and lentil scored a precision value of 0.92 but perfect recall at 1.00. This means that although crops are occasionally misclassified against other classes, all true instances are correctly hit by the model. The small differences in performance could point towards feature engineering similarities among those crops. In all cases, the mirrored high overall scores in addition prove the robust validity of the classifier.

4.4 Confusion Matrix

A confusion matrix, as illustrated in Fig. 4.4, was prepared for the view of visual inspection for predictions by the model. It shows numbers of instances correctly and wrongly classified for each crop. Mostly, the predictions lie along the diagonal line of the matrix, which indicates few misclassifications, having only a few off-diagonal entries, corresponding to a slight dip in performance for jute and lentil. The near-perfect diagonal formation validates the reliability of the predictive scheme implemented by the model.

Training and validation accuracy and loss curves. For real-world assessment, an IoT setup containing the Arduino Uno, NodeMCU, and various sensors was employed. This apparatus was used to collect environmental data: N = 82, P = 77, K = 46, temperature = 32.8°C, humidity = 65%, pH = 3.29, and rainfall = 145 mm. These values were sent as a JSON payload to a Flask web application. The model processed the data and accurately predicted banana to be the suitable crop under those conditions.

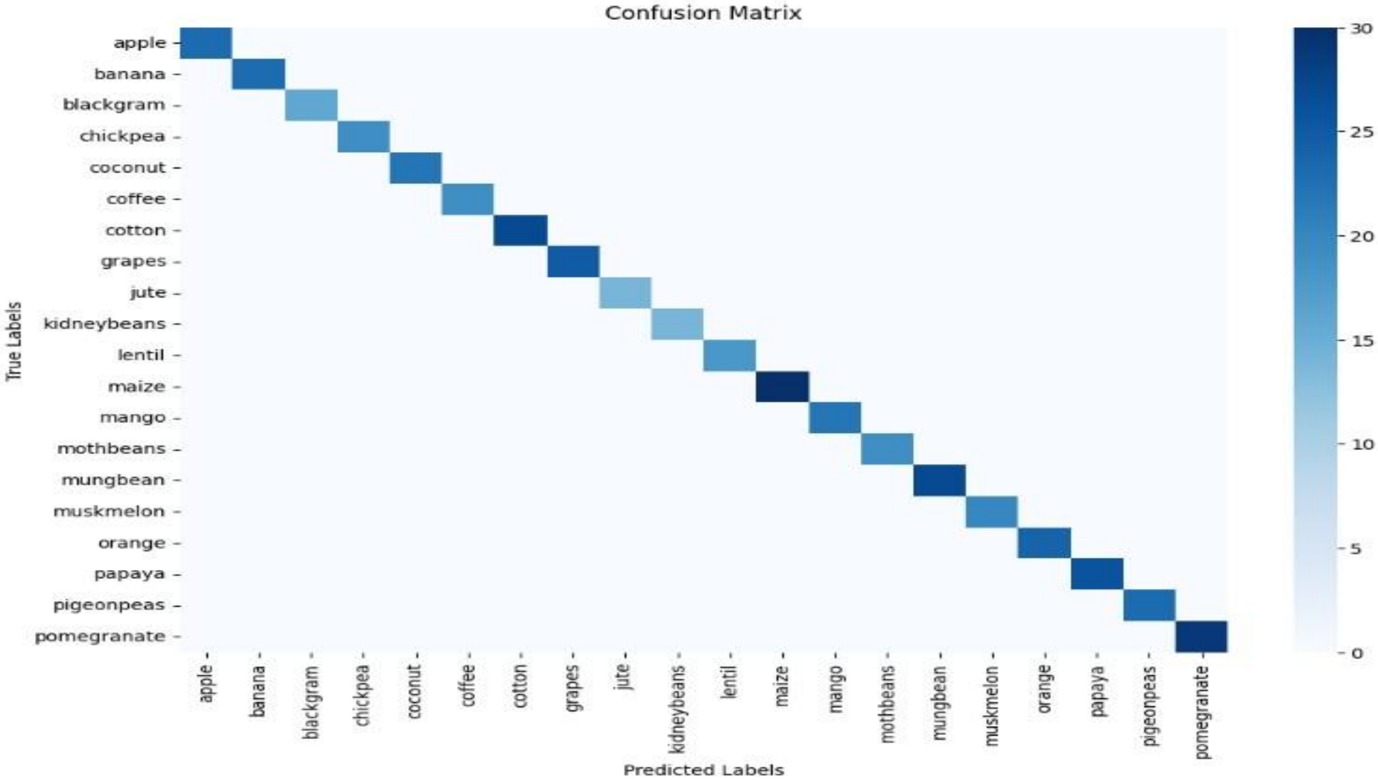

Fig. 4.4 Confusion matrix showing high classification accuracy with minimal misclassification 4.5 Real -Time Sensor-Based Evaluation

4.6 Exploratory Data Visualization

In order to understand and visualize pair-wise relationships, a feature importance plot was generated. To gain further insight into the distribution and interrelationships among the features, a pairplot was used Fig 4.6 This visualization shows the pairwise relations among all the numerical features in the dataset: Nitrogen (N), Phosphorus (P), Potassium (K), temperature, humidity, pH and rainfall. Each subplot shows the scatter plot for two features, while diagonal plots show the kernel density estimation (KDE) of each feature providing a smoothed view of the distribution of data.

The color-coded hues based on crop labels allow for a visual examination of class separability. Distinct clustering patterns are evident in certain feature combinations, such as temperature vs. humidity and Nitrogen vs. Potassium, indicating their potential discriminative power in crop classification. This visualization supports the premise that the selected features contribute meaningfully to differentiating among crop types and reinforces the suitability of the dataset for training a classification model.

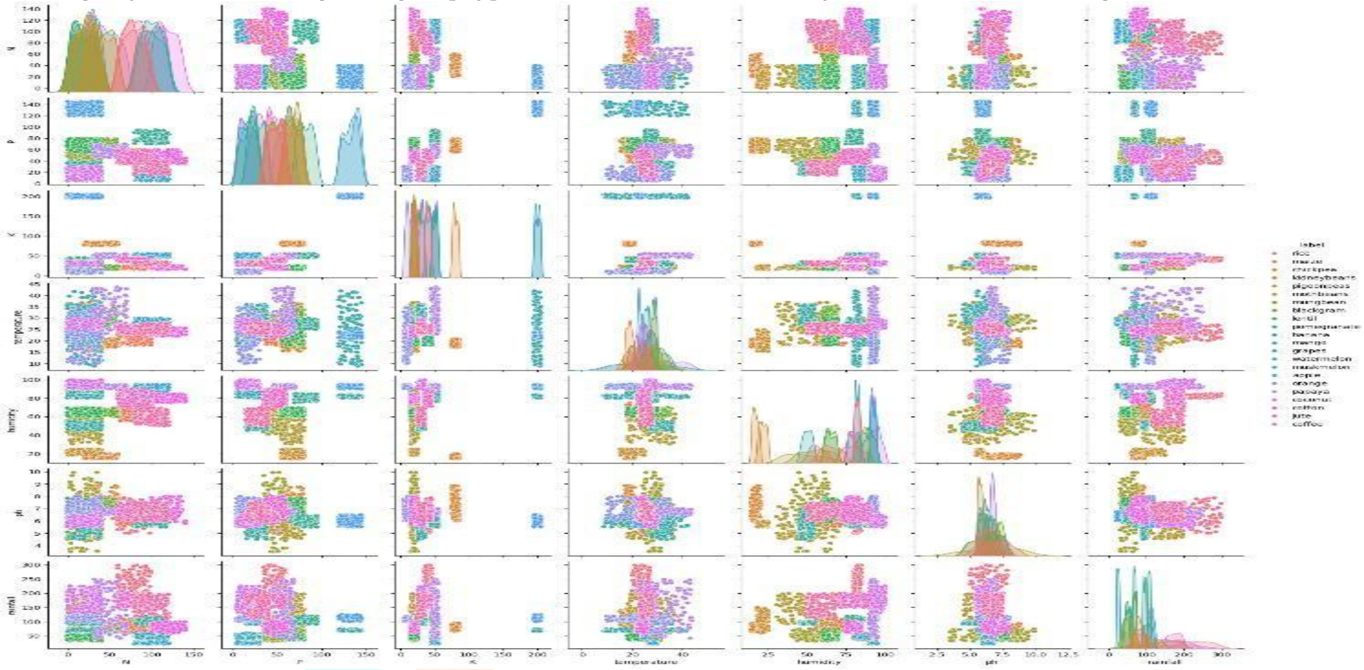

Fig. 4.6 Pairplot showing pairwise feature relationships with KDE distributions on the diagonal. Crop categories are colorcoded, revealing trends and potential clustering that support effective classification.

4.7 Sample Test

These are the instances displayed in Fig. 4.7a and Fig. 4.7b to indicate that real time examples related to the practical implementation of a crop recommendation system. The dashboard is fed with sensor input data such as that of nitrogen, phosphorus, potassium, temperature, humidity, pH, and rainfall from the field. This data is then relayed to the backend, where the trained Random Forest model makes a processing and returns an optimal recommendation along with a confirmation message. Examples above show how conditions would vary with sensor inputs considering high nitrogen and rainfall or moderate potassium and temperature; in such cases, the recommendation has been able to give banana as the most appropriate crop. The dashboard clearly welcomes input parameters and predicted crop, showing practical usability and system reliability in a real-world scenario of precision agriculture.

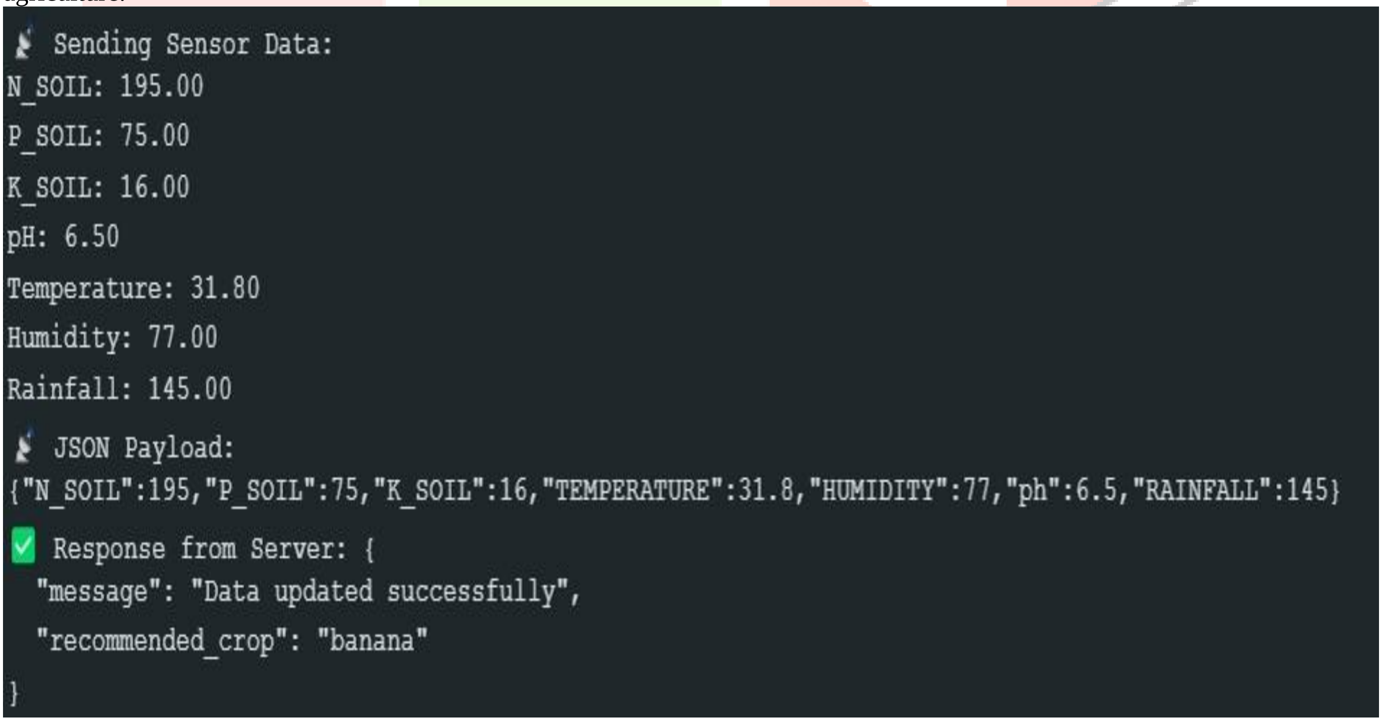

Fig. 4.7a Real-time sensor input data displaying key agricultural parameters which are used by the backend model for crop prediction.

Fig. 4.7b Crop Recommendation Dashboard Interface.

V. Conclusion

This project delivers a robust and practical solution for precision agriculture using a machine learning-based crop recommendation system. By analyzing environmental and soil parameters like nitrogen, phosphorus, potassium, temperature, humidity, pH, and rainfall, it accurately suggests the most suitable crop for specific conditions. The Random Forest classifier, chosen for its accuracy and ability to model complex feature interactions, demonstrated strong performance with high accuracy on unseen data.

Real-time integration with IoT sensors (DHT for temperature and humidity, NPK, and pH sensors) enables dynamic crop recommendations, helping farmers make informed, data-driven decisions. A Flask API ensures smooth communication between the sensors and backend, while a Streamlit dashboard offers an intuitive interface for visualizing sensor readings and crop suggestions. The system is designed with scalability, maintainability, and future enhancements in mind—supporting features like input validation, error handling, multi-language support, and advanced analytics, making it a valuable tool for modern agriculture.

VI. Acknowledgment

The authors express their sincere gratitude to the Department of Electronics and Communication Engineering, KLS Vishwanathrao Deshpande Institute of Technology, Haliyal, for providing the necessary resources and technical support.

References

[1] A. Maazallahi et al., “Naïve Bayes and Random Forest for Crop Yield Prediction,” arXiv preprint arXiv:2404.15392, Apr. 2024**.**

[2] O. Turgut, I. Kok, and S. Ozdemir, “AgroXAI: Explainable AI-Driven Crop Recommendation System for Agriculture 4.0,” arXiv preprint arXiv:2412.16196, Dec. 2024.

[3] N. V., M. S. Josephine, and V. Jeyabalaraja, “IoT-Based Crop Recommendation System for Precision Agriculture,” J. Data Acquis. Process., vol. 39, no. 2, pp. 1–8, Aug. 2024. **[4**] R. Ontigar, V. Ghadi, and S. Bandekar, “IoT-Enabled Crop Recommendation System,” Int. J. Adv. Res. Comput. Commun. Eng., vol. 13, no. 6, pp. 1–5, Jun. 2024.

[5] Prabavathi R; Subha P; Bhuvaneswari M; Prithisha V; Roshini K, “IoT Based Soil pH Detection and Crop Recommendation System,” Int. J. Innov. Sci. Res. Technol., vol. 9, no. 4, Apr. 2024.

[6] Yash Mori and Nandini Modi, "Enhancing OCT Image Classification for Retinal Disease Diagnosis: A Novel Approach Using Squeeze-and-Excitation ResNet," 2024 IEEE 3rd World Conference on Applied Intelligence and Computing (AIC), vol. 1, no. 1, pp. 940–945, Jul. 2024.

[7] B. N. Mohapatra and V. Kale, "Crop Recommendation System Using Machine Learning," ITEGAM-JETIA, vol. 10, no. 48, pp. 1–8, Oct. 2024. **[8]** N. Verma, S. Singh, and A. Joshi, "AI-Based Crop Yield Prediction and Precision Farming Techniques," Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol., vol. 8, no. 2, pp. 213–217, Mar.–Apr. 2023.

[9] A. Pandit, A. Pastapure, A. Bahulikar, P. Gore, and S. A. Deshpande, "IoT Based Crop & Fertilizer Recommendation System Using Machine Learning," Int. J. Res. Appl. Sci. Eng. Technol., vol. 11, no. 4, pp. 1–5, Apr. 2023.

[10] S. Mehta, R. Bansal, and K. Sharma, "AI-Powered Precision Agriculture System for Enhanced Crop Monitoring," Int. J. Adv. Res. Comput. Commun. Eng., vol. 12, no. 5, pp. 112–116, May 2023.

[11] Anusha K, Anusha D J, Sunidhi K R, Trupthi G G, Kiran Kumar K M, "Crop Recommendation Using IoT and Machine Learning," Int. J. Eng. Res. Technol., vol. 11, no. 5, pp. 1–5, Jul. 2023.

[12] P. M. Paithane, "Random Forest Algorithm Use for Crop Recommendation," ITEGAM-JETIA, vol. 9, no. 43, pp. 34–41, Oct. 2023.

[13] K. Bakthavatchalam et al., "IoT Framework for Measurement and Precision Agriculture: Predicting the Crop Using Machine Learning Algorithms," Technologies, vol. 10, no. 1, p. 13, Jan. 2022.

[14] M. Thomas and V. Krishnan, "Implementing AI in Precision Agriculture: A Data-Driven Approach," J. Artif. Intell. Syst., vol. 10, no. 4, pp. 301–307, Nov. 2022.

[15] Y. Reddy and M. Kale, "AI and IoT Integration for Smart Agriculture Applications," Int. J. Comput. Trends Technol., vol. 70, no. 6, pp. 120–125, Dec. 2022.

[16] P. M. Paithane, G. Pruthviraj, J. Atharva, K. Atharv, and P. Yash, "An Approach Towards Crop Recommendation System Using Random Forest Machine Learning Algorithm - A Review," Int. J. Prog. Res. Sci. Eng., vol. 3, no. 12, pp. 117–120, Dec. 2022.

[17] T. Kulkarni, P. Deshmukh, and A. Iyer, "Smart Farming with AI-Powered Decision Systems," Int. J. Agric. Technol. Appl. Sci., vol. 9, no. 1, pp. 47–52, Apr. 2021

[18] D. P. Parikh, J. Jain, T. Gupta, and R. H. Dabhade, "Machine Learning Based Crop Recommendation System," Int. J. Adv. Res. Sci. Commun. Technol., vol. 1, pp. 891–897, Jun. 2021.

[19] D. Gosai, C. Raval, R. Nayak, H. Jayswal, and A. Patel, "Crop Recommendation System using Machine Learning," Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol., vol. 7, no. 3, pp. 554–557, May–Jun. 2021.

[20] R. Sen and M. Bhattacharya, "A Machine Learning Approach for Precision Agriculture," Int. J. Innov. Sci. Eng. Technol., vol. 7, no. 2, pp. 198–202, Feb. 2020.

[21] A. Tatapudi and P. S. Varma, "IoT Based Smart Agriculture Management System Using Machine Learning Classification Algorithms," Int. J. Adv. Sci. Technol., vol. 29, no. 3, pp. 10122–10132, Mar. 2020.

[22] P. Bandara, T. Weerasooriya, and R. T. H., "Crop Recommendation System," Int. J. Comput. Appl., vol. 175, no. 22, pp. 22– 25, Oct. 2020.