International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 09 | Sep 2026 www.irjet.net p-ISSN: 2395-0072

FarmSathi: An Ai-Based Multilingual Farmer Query Support And Agricultural Advisory System

Mr.P. Ramesh[1], V. Jhansi[2], P. Sri Vidya[3], S. Manjunadh[4], S. Pritesh Varma[5]

[1]Professor, Dept. of Computer Science Engineering, Hyderabad Institute of Technology and Management, Telangana, India

[2]Student, Dept. of Computer Science Engineering, Hyderabad Institute of Technology and Management, Telangana, India

[3]Student, Dept, of Computer Science Engineering, Hyderabad Institute of Technology and Management, Telangana, India

[4]Student, Dept, of Computer Science Engineering, Hyderabad Institute of Technology and Management, Telangana, India

[5]Student, Dept, of Computer Science Engineering, Hyderabad Institute of Technology and Management, Telangana, India

---***---

Abstract-*Agriculture plays an important role in supporting livelihoods and ensuring food production. However, farmers often face difficulties in obtaining timely, reliable, and easy-to-understand information for making agricultural decisions. FarmSathi is an AI-based multilingual farmer support and agricultural advisory system developed to provide multiple agricultural services through a single platform. The system integrates crop recommendation, crop disease detection, weather information, and natural language query support to assist farmers with diverse agricultural needs. The system is implemented using a React-based frontend, FastAPI backend, and PostgreSQL database, with machine learning modules integrated into the backend. The crop recommendation module uses seven soil and climatic parameters, namely nitrogen, phosphorus, potassium, temperature, humidity, soil pH, and rainfall. Four machine learning algorithms, namely Decision Tree, Random Forest, Gradient Boosting, and XGBoost, were evaluated for crop recommendation. Random Forest achieved the highest test accuracy of 99.545% on a dataset containing 2200 samples from 22 crop classes. The system also provides image-based crop disease detection, natural language query support for crop management, irrigation, fertilizers, pests, and diseases, weather information based on location, and multilingual interaction. FarmSathi provides a unified platform for accessing agricultural information and supports farmers in making informed, data-driven decisions.*

Key Words: Agricultural Advisory, Crop Recommendation, Random Forest, Crop Disease Detection, Farmer Query Support, Machine Learning, Multilingual Agriculture, Weather-Based Advisory, Natural Language Processing, Smart Farming

1. INTRODUCTION

1.1 Background

Agriculture plays an important role in the lives of farmers and in ensuring food production. Farmers need timely information about crops, soil, irrigation, diseases, pests, and weather, but finding the right information from different sources can be difficult. FarmSathi uses AI and machine learning to bring important agricultural services, such as crop recommendation, disease detection, weather information, and farmer query support, together on a single platform.

1.2 Problem Statement

Farmers face difficulties in obtaining reliable and understandable information for different agricultural problems. Existing resources are often fragmented, requiring farmers to depend on multiple sources for crop selection, disease identification, weather information, and crop management advice. There is a need for a unified and farmer friendly system that can handle diverse agricultural queries and provide appropriate data-driven assistance.

1.3 Objectives

The main objective of FarmSathi:

- To provide natural language support and multilingual interaction for diverse agricultural queries.
- To recommend suitable crops using soil and climatic parameters.
- To provide image-based crop disease detection.
- To provide relevant weather information for agricultural decisions.

- To integrate these services through a React frontend, FastAPI backend and PostgreSQL database.

2. LITERATURE SURVEY

Artificial intelligence and machine learning are increasingly being used in agriculture to help farmers with crop selection, disease identification, and decision making. Earlier studies mainly focused on individual agricultural tasks, while recent research has started exploring integrated and multilingual solutions.

Crop recommendation has been studied using machine learning models based on soil and climatic conditions. An IEEE study developed a crop recommendation system using parameters such as nitrogen, phosphorus, potassium, humidity, temperature, and soil pH [1]. Similarly, Prity et al. compared nine machine learning algorithms for crop recommendation and found Random Forest to be the best-performing model, achieving an accuracy of 99.31% [2]. These studies show that machine learning can be useful for recommending crops based on field conditions.

Plant disease detection is another important application of AI in agriculture. Ferentinos developed deep learning models for detecting plant diseases from leaf images using 87,848 images covering 25 plant species and 58 disease classes. The best-performing model achieved 99.53% accuracy on unseen images [3]. Abade et al. reviewed 121 studies on CNN-based plant disease recognition and found that deep learning has shown strong potential for image-based disease classification. However, challenges such as dataset limitations and differences between controlled images and real field conditions still remain [4].

Recent studies have also explored combining multiple agricultural services into a single system. Singh et al. proposed a multilingual smart farming system that combines crop recommendation, fertilizer recommendation, plant disease detection, and multilingual support [5]. This shows the potential of integrating different AI-based services to provide more comprehensive support to farmers.

Conversational systems are also being explored for agricultural advisory. A recent IEEE study proposed a multilingual conversational recommendation system that enables farmers to interact with an agricultural advisory system through written and spoken communication [6]. Such systems can help address language barriers and provide farmers with easier access to agricultural information.

Overall, existing research shows that AI and machine learning can effectively support crop recommendation, disease detection, and agricultural advisory. However, many existing systems focus on specific agricultural tasks or provide only a limited combination of services. This creates an opportunity for a unified farmer-oriented platform that supports diverse agricultural queries while combining crop recommendation, disease detection, weather information, and multilingual interaction. FarmSathi addresses this need by bringing these agricultural support functions together in a single platform.

3. SYSTEM ARCHITECTURE AND METHODOLOGY

3.1 Architecture

FarmSathi follows a modular architecture in which the React-based frontend, FastAPI backend, PostgreSQL database, external weather service, and AI-based modules work together to provide agricultural assistance. The farmer can interact with the system through text, voice, or image inputs. The React frontend collects the required input and sends the request to the FastAPI backend through application programming interfaces.

The FastAPI backend acts as the main communication layer of the system. It validates the received input and routes the request to the appropriate service. The crop recommendation module uses a Random Forest classifier to recommend suitable crops based on soil and climatic conditions. The disease detection module analyses uploaded crop images to identify possible diseases. The query advisory module handles general agricultural questions related to crop management, irrigation, fertilizers, pests, diseases, and other farming practices. Weather information is obtained using location-based data, while PostgreSQL stores relevant user and application data.

The processed result is returned through the backend and displayed to the farmer through the frontend. This architecture allows the different agricultural services to operate as separate modules while remaining connected through a common platform.

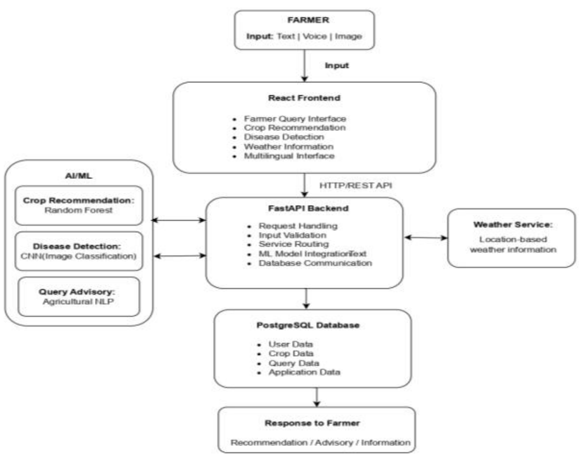

Fig -1: Overall Architecture of FarmSathi System

3.2 Disease Detection Methodology

The disease detection component accepts an image of an affected crop or leaf through the application. The uploaded image is preprocessed operations such as resizing and normalization before classification. An image classification model is used to analyze visual characteristics and identify the corresponding disease category. The detected disease information is then presented to the farmer along with relevant agricultural guidance.

3.3 Crop Recommendation Methodology

The crop recommendation module uses seven parameters: nitrogen (N), phosphorus (P), potassium (K), temperature, humidity, soil pH, and rainfall. These parameters represent important soil and climatic conditions that influence crop suitability.

Multiple machine learning classification algorithms were trained and evaluated to determine the most suitable model for crop recommendation. Decision Tree, Random Forest, Gradient Boosting, and XGBoost were evaluated using the same dataset and test set.

Among the evaluated algorithms, the Random Forest achieved the highest test accuracy of 99.55%. Therefore, the Random Forest classifier was selected as the final crop recommendation model for FarmSathi.

3.4 Query Advisory Methodology

The query advisory component allows farmers to submit agricultural questions using natural language. The system processes the submitted query and identifies the relevant agricultural context, such as crop management, irrigation, fertilizer use, pest problems, disease-related concerns, or general farming practices. Based on the identified query context, appropriate agricultural information is returned to the farmer through the application. The component is designed to support diverse queries rather than restricting farmers to predefined questions.

Multilingual Interaction: The system provides a language selection mechanism that allows farmers to interact with the application in their preferred supported language. English is provided as the default language, while additional regional language support improves accessibility for farmers who may be more comfortable using local languages.

Weather Information: The weather component uses the farmer's location to obtain relevant weather conditions through an external weather service. The backend processes the location information and retrieves weather data, which is then presented through the frontend to support agricultural decision making.

3.5 Dataset and Data Preprocessing

The crop recommendation model was developed by evaluating two datasets from Kaggle [7]. The initial dataset contained soil and climatic information but showed significant class imbalance and relatively low prediction performance. Therefore, a second dataset with a more balanced distribution of crop classes was selected.

The selected dataset contains 2,200 samples belonging to 22 crop classes, with each crop class containing 100 samples. It contains seven input features: N, P, K, temperature, humidity, pH, and rainfall.

The dataset was examined for missing values and duplicate records before model training. No missing values or duplicate records were found. The input features were separated from the crop label, and the dataset was divided into training and testing sets using an 80:20 stratified split. This resulted in 1,760 training samples and 440 testing samples.

3.6 Machine Learning Model Training

Four machine learning algorithms were evaluated on the selected dataset: Decision Tree, Random Forest, Gradient Boosting, and XGBoost. Each model was trained using the seven soil and climatic parameters and evaluated using the same testing dataset.

The obtained results are shown in the chart.

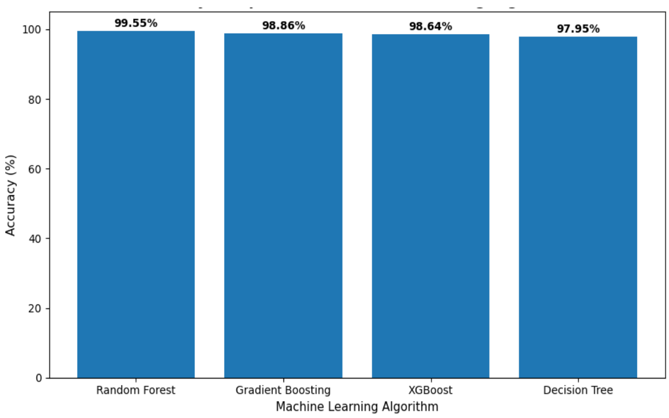

Fig -2: Accuracy Comparison of Machine Learning Algorithms

3.6.1 Random Forest Algorithm

Random Forest is an ensemble machine learning algorithm introduced by Breiman [8] that combines multiple decision trees to perform classification. In FarmSathi, the Random Forest classifier was implemented using the Scikit-learn machine learning library [9]. The model was trained using seven soil and climatic parameters: nitrogen, phosphorus, potassium, temperature, humidity, soil pH, and rainfall. The final model was configured with 200 decision trees and 22 crop classes. The model achieved a test accuracy of 99.55% and was selected as the crop recommendation model for FarmSathi.

Algorithm: Random Forest Based Crop Recommendation

Input: N, P, K, Temperature, Humidity, pH, Rainfall

1. Load the crop recommendation dataset.
2. Separate the seven input features and crop labels.
3. Split the dataset into training and testing sets using an 80:20 stratified split.
4. Initialize the Random Forest classifier with 200 decision trees.
5. Train the Random Forest model using the training data.
6. Obtain the classification output from the trained decision trees.
7. Combine the tree outputs to determine the final crop class.
8. Evaluate the model using the test dataset.
9. Return the recommended crop.

Output: Recommended crop

4. EXPERIMENTATION

4.1 Initial Dataset Experimentation

The first dataset was initially used to train different machine learning models. However, the experimental results showed relatively low performance. Random Forest achieved the highest accuracy among the initial experiments at 51.42%. The dataset also contained considerable class imbalance, with some crop categories having significantly fewer samples than others. Oversampling was subsequently applied to address this issue, but the resulting Random Forest model achieved only 47.55% accuracy.

These results indicated that the initial dataset was not suitable for achieving reliable crop recommendation performance.

Table -1: Initial Dataset Model Results

Model	Accuracy
Decision Tree	39.53%
Random Forest	51.42%
Balanced Random Forest	50.52%
Gradient Boosting	48.84%
XGBoost	49.87%
Oversampled Random Forest	47.55%

4.2 Random Forest Performance

The Random Forest crop recommendation model was trained using 200 decision trees and achieved a test accuracy of 99.55%, with 438 out of 440 test samples correctly classified. Five-fold cross-validation was also performed to examine the consistency of the model. The obtained fold accuracies were 99.55%, 99.32%, 100.00%, 99.77%, and 99.32%, giving a mean accuracy of 99.59% with a standard deviation of 0.27%. These results indicate that the Random Forest model provided consistently high performance across different validation folds.

4.3 Model Evaluation

The final trained model was also tested with a sample set of soil and climatic values:

- N = 90
- P = 42
- K = 43
- Temperature = 25°C
- Humidity = 80%
- pH = 6.5
- Rainfall = 200 mm

The model predicted rice as the most suitable crop. The model's top three predicted class probabilities were rice at 52.00%, jute at 47.50%, and watermelon at 0.50%, with rice receiving the highest predicted probability. This demonstrates how the trained model can process soil and climatic conditions and produce a crop recommendation.

5. IMPLEMENTATION

FarmSathi is implemented using a modular web-based architecture. The frontend is developed using React, which provides the interface through which farmers can enter agricultural information, submit queries, upload crop images, and receive recommendations.

The backend is developed using FastAPI, which handles requests from the frontend, validates inputs, communicates with the machine learning models, and manages communication with the database and other services. PostgreSQL is used for storing application-related information.

The trained Random Forest crop recommendation model is integrated into the backend so that farmer-provided soil and climatic parameters can be processed and converted into crop recommendations. The system architecture also incorporates image-based disease detection, query advisory, weather information, and multilingual interaction to provide a unified agricultural support platform.

6. RESULTS & DISCUSSIONS

The experimental results demonstrate that machine learning can effectively classify suitable crops using soil and climatic parameters. Among the four evaluated algorithms, Random

Forest produced the highest test accuracy of 99.55%. Its performance was also consistent during five-fold cross-validation, where the mean accuracy was 99.59%.

The classification results show that most crop classes were correctly identified. Minor classification errors were observed between a small number of classes, particularly blackgram and maize and rice and jute, which indicates that some crops may have similar ranges of environmental and soil conditions within the dataset.

Overall, the experimental evaluation confirms that the trained Random Forest model provides a strong classification result for the selected crop recommendation dataset. The model is integrated into FarmSathi to generate crop recommendations from the soil and climatic parameters provided by the farmer.

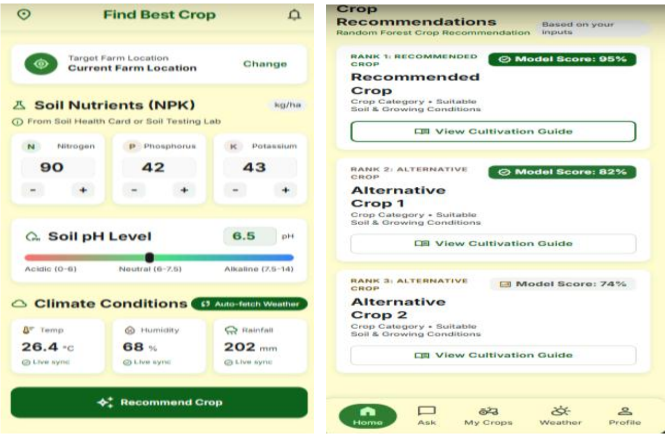

Fig -3: FarmSathi User Interface for Crop Recommendation

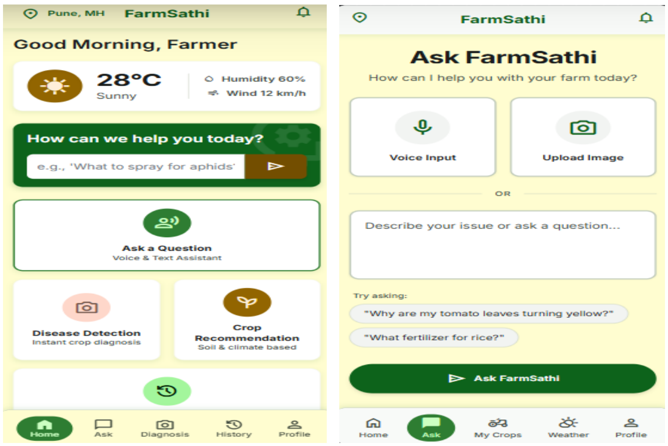

Fig -4: FarmSathi User Interface for Query and Assistance

CONCLUSION

FarmSathi is a unified AI-based platform designed to help farmers make better decisions in their day-to-day farming activities. It brings important services such as crop recommendation, crop disease detection, weather information, and farmer query support together in one place. The crop recommendation module considers soil and climatic factors such as N, P, K, temperature, humidity, pH, and rainfall. Among the machine learning models evaluated, the Random Forest model achieved the highest test accuracy of 99.55%. The disease detection module allows crop images to be analyzed to identify possible diseases, while the query advisory component helps farmers get information related to crop management and other agricultural practices.

By bringing machine learning, image-based disease detection, natural language query support, multilingual interaction, and a centralized backend into one platform, FarmSathi makes agricultural information easier to access and understand. Overall, the system provides a practical approach to using AI and data-driven techniques to support farmers in making informed agricultural decisions.

REFERENCES

1) P. Gokila Brindha, S. ReenaSri, U. Dhanushree, and K. Sivadhanu, "Crop Recommendation Systems using Machine Learning Algorithms," in 2023 International Conference on Recent Advances in Science and Engineering Technology (ICRASET), IEEE, 2023. doi: 10.1109/ICRASET59632.2023.10420164.

2) F. S. Prity, M. M. Hasan, S. H. Saif, M. M. Hossain, S. H. Bhuiyan, M. A. Islam, et al., "Enhancing Agricultural Productivity: A Machine Learning Approach to Crop Recommendations," Human-Centric Intelligent Systems, vol. 4, pp. 497–510, 2024. doi: 10.1007/s44230-024-00081-3.

3) K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," Computers and Electronics in Agriculture, vol. 145, pp. 311–318, 2018. doi: 10.1016/j.compag.2018.01.009.

4) A. Abade, P. A. Ferreira, and F. B. Vidal, "Plant diseases recognition on images using convolutional neural networks: A systematic review," Computers and Electronics in Agriculture, vol. 185, article 106125, 2021. doi: 10.1016/j.compag.2021.106125.

5) S. Singh, H. Dureja, and R. Singh, "A multilingual smart farming system for real time agricultural decisions using machine learning," Discover Applied Sciences, vol. 8, article 613, 2026. doi: 10.1007/s42452-026-08597-y.

6) M. Vijayakumar, Lakshmana Kumar, Y. Abdullah R., and Praveen Kumar NS, "Large Language Model Enabled Multilingual Conversational

Recommendation System for Real Time Agricultural Advisory," in 2026 6th International Conference on Expert Clouds and Applications (ICOECA), IEEE, 2026. doi: 10.1109/ICOECA68095.2026.11485534.

7) A. Ingle, "Crop Recommendation Dataset," Kaggle, [Online].Available:https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset

8) L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001. DOI: 10.1023/A:1010933404324.

9) F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.