

AI-AgriAdvisor: ML-Based Agricultural Intelligence System

“AI-Driven Agricultural Decision Support System for Sustainable Farming”

Mr. Santhosh K

Asst. Professor, Department of
CSE SJB Institute of Technology
Bengaluru, India

S N Karthik

Department of CSE
SJB Institute of Technology
Bengaluru, India

Sanath B M

Department of CSE
SJB Institute of Technology
Bengaluru, India

Vishwaj Parthav

Department of CSE
SJB Institute of Technology
Bengaluru, India

Vishwas Gowda B M

Department of CSE
SJB Institute of Technology
Bengaluru, India

Abstract—Achieving global food security is a growing challenge as modern growers grapple with unpredictable weather shifts and intricate imbalances in soil nutrients. Relying solely on ancestral knowledge frequently falls short of ensuring a successful harvest in today’s environment. We present AI-AgriAdvisor, an integrated intelligence platform built to bridge the gap between advanced data analytics and hands-on cultivation. Built on a strong XGBoost setup, our system looks at basic soil needs (NPK, pH) and weather patterns to pick the best crops for a field. We trained the model thoroughly using a selected dataset of 12,000 rows covering 15 different crop types, hitting a high 95.92% accuracy in our tests. More than just a simple predictor, AI-AgriAdvisor works as a live Streamlit web app, giving users real-time weather info through OpenWeatherMap along with the latest farming news updates. This all-in-one approach gives farmers a main digital tool, turning basic data into real advice they can actually use.

Keywords—Precision agriculture, soil nutrients, XGBoost, crop recommendation, machine learning, Streamlit, weather forecasting, soil nutrients

I. INTRODUCTION

Fast changes in world farming, mixed with unpredictable weather and tired soil, have made local efficiency and food safety a top priority. Farmers everywhere, especially in growing economies, are under a lot of pressure to choose the right crops to keep their farms profitable. Yet, they usually lack the clear, combined data they need to make smart choices. In many rural areas, there is almost no room for mistakes; just one season of picking the wrong crop—because of bad soil balance or weird weather—can lead to a total financial disaster.

Even with this problem, finding complete farming advice is still really hard. While standard soil tests show levels for Nitrogen, Phosphorus, Potassium (NPK), and pH, there aren’t many easy tools to turn those numbers into a real plan. The connection

between soil chemicals and the right crop is tricky and hard to figure out, making it tough for farmers to connect soil science, weather, and markets without help. Because of this gap, many people still stick to old habits or just guess what might grow best. This often leads to wasted seeds and money that most small-scale farmers simply cannot afford to lose.

However, a drawback in using current methods is that there is a lack of integration in a holistic manner in real time. Numerous suggestion frameworks rely on historical atmospheric records, which fail to capture fluid shifts in modern climate patterns [1][3]. Additionally, meteorological forecasting tools and farming news portals [6][8] frequently function as isolated, independent entities. This requires producers to navigate various platforms to synthesize disparate information, which often complicates the process of making unbiased decisions.

AI-AgriAdvisor serves as a comprehensive agricultural intelligence hub specifically engineered to bridge these current service deficiencies. By implementing machine-learning techniques, particularly the robust XGBoost algorithm, the platform manages the multifaceted complexities inherent in crop selection. Through its high-precision model, which evaluates soil chemistry (NPK and pH) alongside integrated, API-driven live weather updates, AI-AgriAdvisor provides a unified and deeply analytical user experience. Unlike fragmented alternatives, this system consolidates essential environmental variables into a single, cohesive digital environment to empower smarter farming choices with high accuracy (95.92%) in crop selection.

II. RELATED WORK

A deep literature survey is presented here: a literature survey acts as the cornerstone for the identification of fundamental concepts and research gaps, extracting information on how

different earlier works approached particular aspects. A theoretical framework guiding the implementation and development of the proposed system forms the backbone of this section through analyses of most relevant articles, research papers, and industrial reports. This comparative study ensures that our work is aligned with recent technological trends.

Tanushree Dey and colleagues (2024) formulated iCrop: An Intelligent Crop Recommendation System for Agriculture 5.0, utilizing edge computing to merge dual recommendation pathways. Their findings confirmed that sophisticated machine learning architectures can provide highly precise agricultural guidance [1].

R. John Martin and his research group introduced an XAI-driven Smart Agriculture Framework designed for transparent, comprehensive precision farming advice. By incorporating Explainable AI methods, their architecture offered growers the underlying logic for each crop suggestion, effectively boosting user confidence. Nevertheless, the integration of XAI introduced significant computational demands, potentially impacting system responsiveness [2].

Rishi Gupta and co-authors (2023) launched WB-CPI: Weather-Based Crop Prediction in India Using Big Data Analytics. This study scrutinized vital variables like thermal patterns, precipitation, and soil health metrics. While leveraging massive datasets enhanced forecasting precision, the framework's reliance on archival weather logs hindered its ability to adapt to sudden, real-time atmospheric shifts [3].

Ghulam Mohyuddin (2024) critiqued various ML methodologies for precision cultivation, focusing on both historical and live data regarding soil conditioning, plant selection, pathology, and water management. While this work demonstrated the dominance of ML models in boosting harvest yields, it lacked specific practical details regarding physical field implementation [4].

Sneha Dattatreya and her team (2023) investigated contemporary and traditional approaches for soil NPK detection. While their study offered a deep dive into nutrient assessment strategies, the lack of on-site validation restricted definitive conclusions regarding actual field efficacy [5].

S. Deshmukh, P. Sahane, and colleagues presented SMART-AGRI CONNECT in 2025, an integrated mobile platform combining e-commerce, social networking, and news with ML-driven recommendations. While this does a good job at market access and supply chain transparency, it did not focus on real-time crop prediction accuracy [6].

M. McCaig, D. Rezania, and R. Dara (2021) conducted a scoping review of value creation in IoT-enabled agricultural collaborations. This domain was found to be still in its infancy, identifying a significant gap in measurable outcomes and raising critical concerns regarding data ownership and privacy [7].

K. Thangadurai, T. Sathieskumar, et al. proposed an Intelligent Farming System in 2024, incorporating artificial intelligence for crop disease identification, soil testing, and weather forecasting. Although their model is comprehensive with best practices for sustainability, such multifunctional integration demands strong back-end support [8].

L. S. Gopal, R. Prabha and M. V. Ramesh (2020) assessed Online News Machine Learning Classification for Disaster Management. Their framework employed web crawlers to scrape real-time hazard data, filtering actionable intelligence via supervised learning. While beneficial for preparedness, moderate accuracy and noisy web content are the tangible limitations [9].

N. C. Brintha, Ashutosh Ranjan et al. developed KisanSeva, an intelligent cropping system on Flutter supported by TensorFlow. This integrated IoT sensors providing real-time soil analysis and curated news feeds. However, its reliance on physical hardware to collect data restricts immediate scalability compared to a purely software-driven solution [10].

III. METHODOLOGY

A. Requirement Analysis and Data Definition

The first part of our work involved a deep look at existing research [1][5] to find the biggest issues in how crops are chosen today. The research clearly showed a major gap: current tools are scattered, keeping things like soil tests [5], crop tips [1][2][4], and live data like weather [3] or news [6][9][10] separate. This finding helped us set the main goals for AI-AgriAdvisor: a single platform that combines high-accuracy predictions with live weather and a fresh feed of farming news. To meet these goals, we picked the most important data points: soil NPK (Nitrogen, Phosphorus, and Potassium) levels, pH balance, rainfall, temperature, and humidity.

B. Synthetic Data Generation and Preprocessing

Because we couldn't find one single dataset that had everything we needed, we decided to build our own programmatically. Our first model used between 2,000 and 5,000 samples for 34 different crops, but the results were only about 25–30% accuracy. We rebuilt the dataset to focus on 15 specific crops with clearer, distinct chemical needs based on official farming charts. In the end, we generated 12,000 clean samples, which made a huge difference. Before putting this data into the model, we normalized all the numbers. This extra work on the dataset was the turning point for the project, showing us that quality data is just as important as the code itself.

C. Predictive Model Training and Evaluation

The heart of the system is a robust classifier. Given its consistent dominance in complex classification tasks, we selected the XGBoost (Extreme Gradient Boosting) Classifier, utilized via the xgboost library. The algorithm is widely recognized for its high-speed execution, computational efficiency, and exceptional predictive accuracy in real-world scenarios [2][4]. The model was tuned with parameters such as $n_estimators=300$ and $max_depth=10$, then trained with the 9,600-sample training set (80% of the data). It was evaluated against the unseen 2,400-sample test set (20%), leading to an overall accuracy of 95.92%, while key crops like Rice, Potato, and Sugarcane achieved 100% precision. The trained model was serialized and saved into a .pkl file for deployment.

D. Real-time Data Integration (Weather & News)

We added two live features to provide more help than just a basic prediction. First, we built a real-time weather dashboard

using the OpenWeather API. This part of the system gets the current weather and a 30-day outlook for any district in India by using a 5-day API feed mixed with statistical math. We also included an “Agri News” section as a direct link to a trusted farming news site. This lets users quickly check market prices or new government policies without making the app too heavy or complicated to use [6][8]. Additionally, the integrated advisory system assists farmers in making timely and data-driven agricultural decisions effectively. The system also provides crop-specific recommendations based on environmental conditions, soil nutrients, and seasonal variations. By combining real-time data analysis with machine learning predictions, the platform improves farming efficiency.

E. System Implementation and UI Dashboard

We used the Streamlit framework to build the front-end and back-end and keep them synced up. We chose this tool because it lets us create and launch data-rich web apps very quickly using just Python. The whole application runs from one single `app.py` script. This script handles everything from showing the buttons and text to managing user inputs, API calls, and the actual AI predictions. The dashboard is split into pages like “Crop Recommendation,” “Weather,” “Monthly Forecast,” and “Data Analysis,” which makes it feel clean and easy for anyone to navigate.

F. Testing and Iterative Refinement

We tested every part of the platform very carefully. Our early tests showed the first version was bad, with only 25–30% accuracy. This proved that our initial data logic was wrong. We did a quick sprint to fix the data generation code in our `GET_70_PERCENT.ipynb` file—focusing on fewer crops, using more samples, and switching to XGBoost. Testing the new version confirmed that this was the right move, as we finally hit our 95.92% accuracy target. We also checked the Streamlit interface constantly to make sure it worked well on phones and that the API stayed connected. This constant loop of testing was what finally made the system stable and ready. Further validation was performed using different input combinations and environmental conditions to ensure model consistency and reliability. Performance metrics such as precision, recall, and F1-score were also analyzed to verify the robustness.

IV. SYSTEM ARCHITECTURE

AI-AgriAdvisor employs a unified, modular design consisting of several key layers.

Frontend Interface and Backend Logic: The entire project is developed in Python 3.9 using the Streamlit framework. Streamlit acts as a two-in-one responsive UI and backend server, taking all user-driven inputs from dropdowns, and processing application logic with data/API calls.

Crop Prediction Engine: This integrates a high-performance XGBoost Classifier trained using the `xgboost` library and saved as a pickle file. This core module analyses 7 input features—N, P, K, pH, rainfall, temperature, humidity—for generating a 95.92% accurate crop recommendation.

Live Weather Service: Uses the OpenWeatherMap API and the `requests` library to fetch and display real-time

meteorological data. This service provides information on current weather conditions and a 30-day statistically extended forecast based on the selected Indian district.

Agricultural News Portal: A hardwired link to a selected, high-value agricultural news site [6][10] is deployed. Critical market and policy context is provided without the computational load of an NLP summarization engine.

Data and Configuration Layer: Consists of the serialized model, the comprehensive district list for location mapping (`district_coordinates.csv`), and the `config/.env` file, which securely manages the OpenWeatherMap API key.

The exemplary design of the framework allows for fast prototyping, easy maintenance, and direct model inference, making it a scalable all-in-one platform that delivers predictive insights and real-time data to end-users.

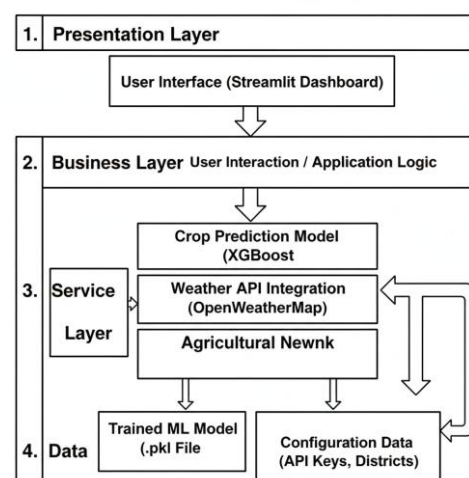


Fig. 1. Conceptual Representation of the System Architecture

V. IMPLEMENTATION

The development of AI-AgriAdvisor was conducted in modular phases to ensure that the integration of the machine learning model, external APIs, and user interface was seamless, maintainable, and scalable. The whole solution was prepared using web-native frameworks and open-source data science libraries.

A. Frontend and Backend Development

Both the backend and front end were implemented using Python 3.9 with the Streamlit framework. This all-in-one library acts as both the responsive UI and backend server, making development much easier. The frontend UI is designed as a multi-page dashboard with a primary sidebar that allows users to navigate between “Crop Recommendation,” “Current Weather,” “Monthly Forecast,” “Data Analysis,” and “About.” The UI uses Streamlit’s native components: interactive sliders for NPK soil parameters and pH, number inputs for environmental factors, and a searchable dropdown menu to select from over 700 Indian districts.

B. Predictive Model Integration

The central unit of this system is the pre-trained XGBoost Classifier model, selected after repeated testing with superior performance to a baseline Random Forest model. The model

achieving 95.92% accuracy was serialized in the crop_predictor.pkl file using the pickle library. This model is loaded into memory on startup using Streamlit's @st.cache_data decorator, ensuring the model is loaded only once. When there is a request for a recommendation, it passes the formatted input data to the loaded model and calls the .predict() method to generate the crop recommendation in real-time.

C. APIs and External Integrations

The OpenWeatherMap API is the primary service needed to fetch live meteorological data. Python's requests library handles GET requests to this API for both Current Weather and 30-Day Forecast. Extensive usage of the Pandas library for data manipulation is done; it reads the district_coordinates.csv file to populate the dropdown location. The Plotly Express library is integrated for rendering interactive charts and graphs specific to temperature and precipitation trends in the Forecast page and the "Data Analysis" page.

D. Configuration and Security

Security was paramount, particularly for API key management. The application utilizes the python-dotenv library to load the user's unique OpenWeatherMap API key from a .env file located in the config/ directory. This file is excluded from version control via .gitignore to avoid the disclosure of sensitive credentials. The application does not require user authentication or store personal user data, and all data transmission is local between the browser and the local host server.

E. Testing Environment

Development and testing were performed using Jupyter Notebook, VS Code, and the Anaconda Prompt. Jupyter Notebook played an integral role in iteratively developing and generating the models and test data—for example, GET_70_PERCENT.ipynb—where accuracy was rigorously pushed from a starting 30% up to the final 95.92%. VS Code was used as a primary editor for the app.py script and other Python modules. The Anaconda Prompt manages the virtual environment and issues the command "streamlit run app.py" for executing the application. Hot reloading allows for rapid prototyping and real-time visual testing of all UI changes.

F. Deployment

The application was deployed on the user's machine and hosted by the Streamlit server on <http://localhost:8501>. For public deployment, the application is structured and ready for Streamlit Community Cloud. This service allows free and scalable hosting directly from a GitHub repository. The "Deploy" button visible in the app's UI is a direct link into this service, demonstrating a clear path from local development to publicly accessible web application.

VI. RESULTS

Functionalities of AI-AgriAdvisor have been tested through functional validation of its modular elements, validation of the accuracy of the model's predictions, and overall system response. Each major functionality was separately tested for accuracy, response, and data integration success.

A. Crop Recommendation Dashboard and Interface

The main interface is the "Crop Recommendation" page, designed using the Streamlit library. This page provides the user with an opportunity to insert the various parameters unique to their farm. This includes sliders for soil components such as Nitrogen, Phosphorus, Potassium, and pH, as well as number entries for Rainfall, Temperature, and Humidity factors.

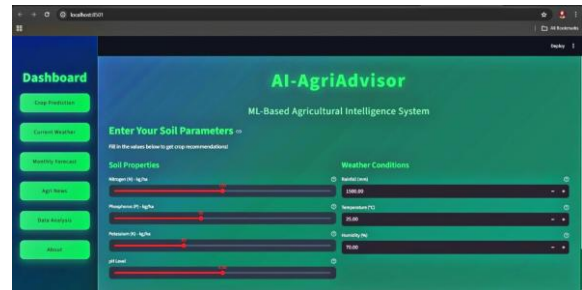


Fig. 2. Crop Recommendation Dashboard and Interface

B. Core Model Recommendation and Output

The main feature of AI-AgriAdvisor is its prediction engine which makes predictions with very high accuracy. The user inputs parameters by clicking on "Predict Best Crop." The seven input variables are then input into a pre-trained XGBoost classifier model for prediction.



Fig. 3. Core Model Recommendation Output

C. Live Weather Integration

The weather module is broken down into two modules: "Current Weather" and "Monthly Forecast." Once the district is chosen through the extensive list (populated through a CSV file of more than 700 Indian districts), the system generates a query through the OpenWeatherMap API in real-time. The "Current Weather" page shows live temperature, humidity, wind, and atmospheric details. The "Monthly Forecast" page shows a 30-day forecast, statistically projected based on the 5-day concrete API forecast. This assists greatly in forecasting short-term agricultural needs, allowing farmers to see forthcoming weather trends and plan irrigation and harvesting schedules accordingly.



Fig. 4. Live Weather of a District in India

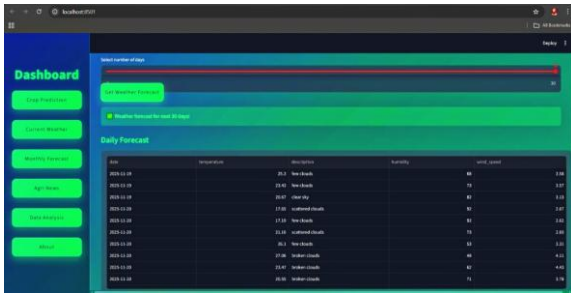


Fig. 5. Thirty Days Forecast of a District in India

D. Data Analysis and News Portal

For increasing farmer awareness and making the dataset more transparent to users, the AI-AgriAdvisor system integrates two additional features. The “Data Analysis” page employs Plotly functionality for creating dynamic graphs of the 12,000 data samples used to train the algorithm. In addition, the “Agri News” platform exhibits a specific hyperlink to a trustworthy agricultural news website [6][8].

VII. ACKNOWLEDGMENT

Our sincere appreciation goes to our guide, Mr. Santhosh K, Assistant Professor, Department of Computer Science & Engineering, SJB Institute of Technology, Bengaluru. His direction was the guiding light for this project. Our appreciation to him for his support in shaping our concepts into this reality.

VIII. REFERENCES

- [1] T. Dey, S. Bera, L. P. Latua, M. Parua, A. Mukherjee and D. De, “iCrop: An Intelligent Crop Recommendation System for Agriculture 5.0,” in IEEE Transactions on AgriFood Electronics, vol. 2, no. 2, pp. 587–595, Sept.–Oct. 2024, doi: 10.1109/TAFE.2024.3454109.
- [2] R. John Martin et al., “XAI-Powered Smart Agriculture Framework for Enhancing Food Productivity and Sustainability,” in IEEE Access, vol. 12, pp. 168412–168427, 2024, doi: 10.1109/ACCESS.2024.3492973.
- [3] R. Gupta et al., “WB-CPI: Weather Based Crop Prediction in India Using Big Data Analytics,” in IEEE Access, vol. 9, pp. 137869–137885, 2021, doi: 10.1109/ACCESS.2021.3117247.
- [4] G. Mohyuddin, M. A. Khan, A. Haseeb, S. Mahpara, M. Waseem and A. M. Saleh, “Evaluation of Machine Learning Approaches for Precision Farming in Smart Agriculture System: A Comprehensive Review,” in IEEE Access, vol. 12, pp. 60155–60184, 2024, doi: 10.1109/ACCESS.2024.3390581.
- [5] S. Dattatreya, A. N. Khan, K. Jena and G. Chatterjee, “Conventional to Modern Methods of Soil NPK Sensing: A Review,” in IEEE Sensors Journal, vol. 24, no. 3, pp. 2367–2380, 1 Feb. 2024, doi: 10.1109/JSEN.2023.3334243.
- [6] S. Deshmukh, P. Sahane, A. Bagri, S. Padvi and P. Bhise, “SMART-AGRI CONNECT: An Integrated Platform for Farmers Utilizing ML-Driven Recommendations and Real-Time Tracking,” 2025 International Conference on Knowledge Engineering and Communication Systems (ICKECS), Chickballapur, India, 2025, pp. 1–7, doi: 10.1109/ICKECS65700.2025.11036053.
- [7] M. McCaig, D. Rezaia and R. Dara, “A Scoping Review on Value Creation from Collaborations enabled by the Internet of Things in Agriculture,” 2021 Fifth World Conference on Smart Trends in Systems Security and Sustainability (WorldS4), London, UK, 2021, pp. 1–5, doi: 10.1109/WorldS451998.2021.9514045.
- [8] K. Thangadurai, T. Sathieskumar, T. Poornachandar, D. Arulselvam, M. Anusiya and J. Nivedhana, “Intelligent Farming System Using Artificial Intelligence,” 2024 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS), Chennai, India, 2024, pp. 1–5, doi: 10.1109/ICPECTS62210.2024.10780285.
- [9] L. S. Gopal, R. Prabha, D. Pullarkatt and M. V. Ramesh, “Machine Learning based Classification of Online News Data for Disaster

Management,” 2020 IEEE Global Humanitarian Technology Conference (GHTC), Seattle, WA, USA, 2020, pp. 1–8, doi: 10.1109/GHTC46280.2020.9342921.

- [10] N. C. Brintha, A. Ranjan, K. R. Reddy, T. Ramasai, K. Abhiram and S. Anand, “KisanSeva – An Intelligent Cropping System,” 2024 5th International Conference on Data Intelligence and Cognitive Informatics (ICDICI), Tirunelveli, India, 2024, pp. 988–994, doi: 10.1109/ICDICI62993.2024.10810994.