IJCRT.ORG ISSN : 2320-2882
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

Development Of An Integrated Full-Stack Smart Agriculture System For Real-Time Farm Monitoring And Decision Making

[1]Dr. Sumitha C, [2]SHASHANK C M, [3]MANMOHAN G K, [4] KARTHIK JOGI K, [5]BHARATHRAJ GOWDA J B

[1] Associate Professor and Hod, Dept. of Computer Science and Engineering, G. Madegowda Institute of Technology, Bharathinagara, Karnataka, India.
[2] Student, Dept. of Computer Science and Engineering, G. Madegowda Institute of Technology, Bharathinagara, Karnataka, India.
[3] Student, Dept. of Computer Science and Engineering, G. Madegowda Institute of Technology, Bharathinagara, Karnataka, India.
[4] Student, Dept. of Computer Science and Engineering, G. Madegowda Institute of Technology, Bharathinagara, Karnataka, India.
[5] Student, Dept. of Computer Science and Engineering, G. Madegowda Institute of Technology, Bharathinagara, Karnataka, India.

Abstract: This project presents an integrated full-stack Smart Agriculture Decision Support System that delivers AI-driven crop advisory and digital services to farmers through a unified web platform. The system combines an Artificial Neural Network (ANN)–based crop recommendation module with a Convolutional Neural Network (CNN)–based plant disease detection module that identifies diseases from leaf images and suggests suitable control measures. Alongside these core analytics, the portal offers plant identification, a farmer marketplace, curated government scheme information, agriculture news aggregation, and an AI-powered chatbot (Agri-Bot) for natural-language query support.

The application is implemented using a modular architecture with a responsive front end and RESTful back end, enabling smooth integration of machine-learning models and external APIs such as weather services. Functional testing and preliminary user evaluation indicate that the platform provides accurate recommendations and an intuitive user experience, reducing the effort required for farmers to access reliable information scattered across multiple sources. The proposed system demonstrates the feasibility of a one-stop smart farming portal and provides a scalable foundation for future enhancements such as IoT sensor integration, multilingual voice interaction, and region-specific model customization.

I. Introduction

Agriculture is one of the most critical sectors of the Indian economy, providing livelihood to a large share of the population and supplying food and raw materials to multiple industries. Despite its importance, farmers still depend heavily on experience, local advice, and manual observation to make crucial decisions such as crop selection, disease management, and marketing. With increasing climate variability, fluctuating market prices, and complex government policies, these traditional practices are often inadequate. The rapid growth of internet connectivity and affordable smartphones in rural areas creates an opportunity to deliver intelligent, data-driven decision support directly to farmers through full-stack web applications.

Agriculture has always been a critical pillar of the Indian economy, providing food security, employment, and raw material for several industries. Rural livelihoods, cultural practices, and regional economies are all deeply connected to agricultural productivity and stability. Yet the sector is facing unprecedented pressure due to climate change, fragmented land holdings, fluctuating market prices, and limited access to reliable information.

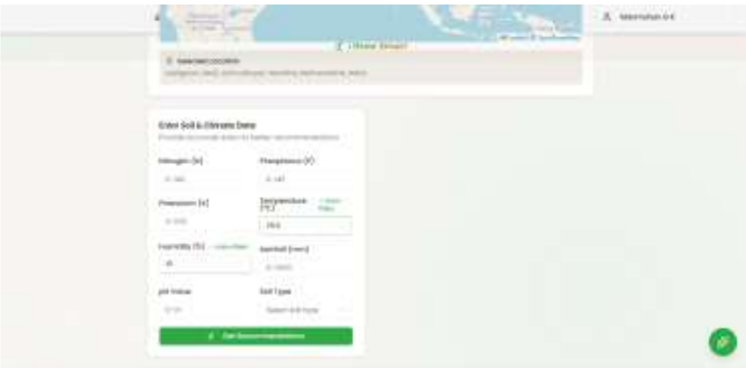
Agriculture is one of the most fundamental sectors supporting human survival and global economic development. It provides food, raw materials, and employment to a significant portion of the world's population. In developing countries like India, agriculture remains the backbone of the economy, directly or indirectly supporting millions of livelihoods. However, despite its importance, the agricultural sector faces numerous challenges that threaten productivity, sustainability, and farmer income.

II.METHODOLOGY

The methodology of this project follows a data-driven engineering approach that combines machine-learning model development with full-stack web application design to build an integrated Smart Agriculture Decision Support System. First, a detailed problem analysis and requirement study identified the need for crop recommendation, plant disease detection, plant identification, a marketplace, scheme and news information, and an AI-based chatbot. Tabular datasets containing soil nutrients (N, P, K), pH, temperature, humidity, rainfall and corresponding crops, along with leaf-image datasets for major crop diseases, were collected from open sources, cleaned, normalised, augmented and split into training and testing sets. An Artificial Neural Network (ANN) was then designed and tuned to recommend suitable crops based on environmental and soil parameters, while a Convolutional Neural Network (CNN) was implemented to classify leaf images into healthy and diseased classes; both models were evaluated using accuracy and confusion matrices and the best versions were exported for deployment. A three-tier architecture with a responsive web front end, RESTful back end and database was developed, organising the system into independent modules (crop recommendation, disease detection, plant identification, marketplace, schemes and news, and Agri-Bot chatbot) that communicate via APIs. The trained ANN and CNN models were integrated into the back end so that user inputs (soil values or leaf images) are preprocessed, passed to the appropriate model, and the predictions are returned as advisory messages within the UI or chatbot conversation. Finally, the system underwent unit, integration and system testing across all workflows, feedback from trial users was used to refine the interfaces and responses, and the complete design, implementation and evaluation were documented as part of the project report.

1. Data Acquisition

- Focus on functional correctness of each module so that all services (ANN crop recommendation, CNN disease detection, marketplace, schemes, news, chatbot) behave exactly as specified.
- Ensure high ML accuracy using metrics like accuracy, precision, recall, F1-score and confusion matrix on separate test datasets.
- Maintain good performance/response time so predictions, page loads and chatbot replies are fast even on low-end devices and slow networks.
- Prioritise usability with simple navigation, clear labels, local language support and minimal input steps for farmers.
- Guarantee reliability and availability by handling errors gracefully, avoiding crashes and keeping the system accessible during peak use.
- Provide security for logins, user data and uploaded images through authentication, authorization and safe storage practices.
- Design for maintainability and scalability using modular architecture so new crops, diseases, features and languages can be added easily.
- Achieve good portability via responsive web design and server-agnostic deployment, allowing access from mobiles, tablets and desktops.



ANN Algorithm – Working Principle

Input layer:

The input layer receives the numerical features describing a farmer's field, such as nitrogen (N), phosphorus (P), potassium (K), soil pH, moisture, average temperature, rainfall, and season code. Each feature correspondsto one input neuron.

Output layer:

The output layer contains one neuron for each supported crop. The output of each neuron represents a suitability score or probability indicating how appropriate that crop is for the given conditions.

2. Forward propagation

Each input feature xi is multiplied by a corresponding weight wi and summed with a bias term b to obtain for each neuron.

$$z = \sum_i w_i x_i + b$$

The activation function f(·) is applied to this weighted sum to produce the neuron's output a = f(z). Common activation functions are ReLU, sigmoid, and tanh.

The outputs from the input layer are fed into the first hidden layer, then successively into deeper hidden layers, and finally into the output layer.

The final output is a vector of scores $y = [y_1, y_2, \ldots, y_k]$, where y_j indicates the suitability of crop j. These scores can be normalised to represent probabilities and ranked to obtain the top crop recommendations.

3.Training the ANN using backpropagation

Before deployment, the ANN must be trained on historical data linking soil and climate conditions to successful crops. The training process uses the backpropagation algorithm with gradient descent.

1. Initialisation:

Network weights and biases are initialised to small random values.

2. Forward pass:

For each training example, the network computes outputs using the current weights (as described in the previous subsection).

3. Loss calculation:

A loss function L (e.g., mean squared error or cross-entropy) measures the difference between the network's output and the true target label representing the correct crop.

4. Backpropagation:

The gradient of the loss with respect to each weight $\partial L/\partial w$ is computed by propagating error signals backwards through the network layers. This step determines how much each weight contributed to the overall error.

5. Weightupdate

Weights are updated using gradient descent where η is the learning rate controlling the step size of updates.

$$w_{\text{new}} = w_{\text{old}} - \eta \cdot \frac{\partial L}{\partial w}$$

III. Important Algorithms in the Project (Simple Explanation)

1. ANN – Artificial Neural Network (for Crop Recommendation)
- Takes inputs like N, P, K, pH, moisture, temperature and rainfall.
- Learns from past data which crop grows best in which condition.
- When the farmer enters soil and weather values, ANN gives the best suitable crop.
2. CNN – Convolutional Neural Network (for Disease Detection)
- Works on leaf images taken by the farmer.
- Automatically looks for spots, colour changes and texture on the leaf.
- Outputs the disease name or says the plant is healthy.
- Runs in the browser using TensorFlow.js, so results come in real time.
3. CNN for Crop Price Prediction
- Uses past market prices, demand and season data.
- Learns patterns in the price history.
- Predicts the future price range so farmers know when it is better to sell.
4. Naïve Bayes (only in literature review)
- A simple probability-based algorithm used in old research for crop prediction.
- Assumes all features are independent, so it is less accurate for complex farm data.
- In this project it is only discussed, not used; we use ANN instead.
5. AI Chatbot with NLP
- Understands farmers' questions in normal language.
- Detects the intent (crop advice, disease doubt, schemes, prices).
- Calls the right module (ANN, CNN, schemes, etc.) and sends a simple reply.
- Can be extended later with Kannada text/voice support.

3. Feature Engineering

Feature engineering was done to optimize interpretations. The model followed these steps:

- Agronomic and climatic feature selection

Soil and weather attributes such as N, P, K, pH, moisture, temperature, humidity, rainfall and season were examined for their relevance to crop yield and suitability. Highly correlated or redundant attributes were reduced so that the ANN focuses on the most informative factors.

- Encoding and scaling

Categorical attributes (season, crop labels, disease classes) were converted into numerical form using one-hot encoding. All continuous features were normalised or standardised so that the ANN and CNN models converge faster and are not biased towards parameters with larger numeric ranges.

- Image preprocessing for CNN

Leaf images were resized to a fixed resolution, converted to a consistent colour space, and normalised pixel-wise. Data augmentation operations such as rotation, flipping and zooming were used to increase variation and reduce overfitting in the disease-detection model.

Derived features and external data

Additional features such as heat-index-like combinations of temperature and humidity, simple season codes, and market-side attributes (price trends, demand indicators) were introduced to support better crop and price recommendations.

- Development of Machine Learning Models

Multiple supervised learning models were developed and evaluated to identify the most suitable approach for each module of the system.

- Crop recommendation models

Classical models such as decision trees, random forest and Naïve Bayes were first implemented as baselines. These were then compared against a feed-forward Artificial Neural Network trained on the engineered agronomic features. The dataset was typically split in an 80:20 or 70:30 ratio for training and testing, and hyperparameters (hidden layers, neurons, learning rate, batch size) were tuned using grid- or random-search strategies. The ANN showed better performance on non-linear relationships and was selected for deployment.

- Plant disease detection models

For image-based classification, different CNN configurations were tested, varying the number of convolution–pooling blocks, kernel sizes, dropout rates and fully connected layers. Transfer-learning experiments using pre-trained CNN backbones were also considered as references. The final CNN architecture was chosen based on accuracy, F1-score and generalisation behaviour on unseen leaf images.

- Price prediction and auxiliary models

For price prediction, deep models capable of capturing temporal patterns (e.g., CNN-based sequence model) were trained on historical price data. Performance was evaluated using regression metrics such as RMSE and MAE. Across modules, ensemble baselines like random forest helped confirm that the chosen neural models indeed offered better handling of noise and non-linearities.

- System Integration and Deployment

The best-performing models were integrated into a full-stack smart agriculture application for real-time advisory and interaction.

- Backend processing (Python + ML models)

Trained ANN and CNN models are exported and loaded inside a Python-based backend service. This layer exposes endpoints for crop recommendation, disease detection, price prediction and chatbot queries, handling preprocessing, model inference and post-processing of results.

- Front-end interface (Web application)

A responsive web interface allows farmers to enter soil parameters, upload leaf images, view recommendations, explore schemes and news, and interact with the AgriBot chatbot. Separate pages and components are provided for each module, while maintaining a consistent design language.

- Data and service pipeline

The application connects to external services such as weather APIs and, where available, market-price feeds. Incoming data is cleaned and formatted before being passed to the models for instant prediction. User requests and model outputs are stored for logging and potential future retraining.

- Visualisation and guidance

Outputs are presented through clear visual elements such as cards, charts and status labels. Farmers can see recommended crops, predicted diseases with remedy suggestions, indicative price trends, and contextual tips or alerts generated by the chatbot and rules engine.

IV. MODELING

Modeling
- Crop recommendation model (ANN)
- Input features: N, P, K, pH, soil moisture, temperature, rainfall, season.
- Model type: feed-forward Artificial Neural Network trained on labelled crop–soil–climate data.
- Data split into training and testing sets to avoid overfitting.
- Disease detection model (CNN)
- Input: preprocessed leaf images (resized, normalised, augmented).
- Model type: Convolutional Neural Network with convolution, pooling and fully connected layers for multi-class disease classification.
- Price prediction model (CNN / deep model)
- Input: historical price time-series with seasonal and demand information.
- Model type: CNN-based sequence model that learns local patterns and trends for forecasting future prices, similar to other AI-based agricultural DSS models.

Analysis
- Evaluation metrics
- For ANN and CNN models: accuracy, precision, recall and F1-score; confusion matrices used to inspect misclassifications and class imbalance.
- For regression-style price prediction: error metrics such as RMSE or MAE and coefficient of determination R^2 as used in other agricultural ML studies.
- Experimental setup
- Data split into training, validation and test sets; hyperparameters tuned using validation performance.
- Comparison with simpler baselines from literature (e.g., Naïve Bayes, decision trees, rule-based systems) to show improvement in accuracy and robustness.
- System-level analysis
- Response time of key APIs (crop recommendation, disease detection, chatbot) measured to ensure practical use in the field.
- Usability and acceptability checked through trial users, aligning with best practices for smart agriculture decision support systems.

V. RESULTS AND DISCUSSION

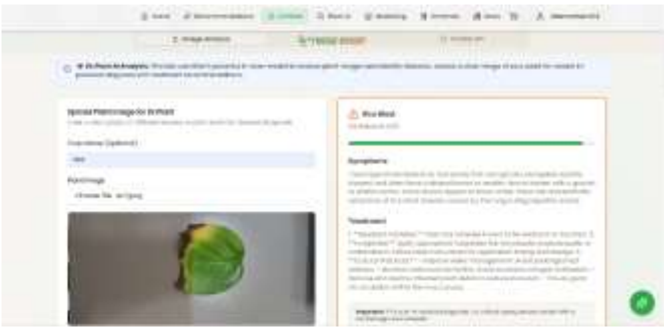
In this project, the Integrated Smart Agriculture System was successfully implemented as a working full-stack application that combines web technologies with ANN and CNN models for decision support.
System and integration testing confirmed that end-to-end workflows (for example, farmer login → recommendation → disease detection → listing creation → buyer order → scheme search) executed correctly, with response times staying within the defined non-functional limits on the test server. Trial users (students/faculty) were able to complete typical tasks without assistance and reported that having recommendations, disease diagnosis, schemes, and marketplace in one interface was more convenient than using multiple separate applications.

- crop recommendation

This interface is the front-end entry point for the ANN algorithm described in your report: the farmer provides soil nutrients and climate data, the system automatically fills location-based climate values (temperature and humidity), and upon clicking the button the ANN computes crop suitability and sends the results back to be displayed on the next screen. This design is consistent with modern smart-farming tools that use structured forms plus auto-fetched weather data to drive crop recommendation models.

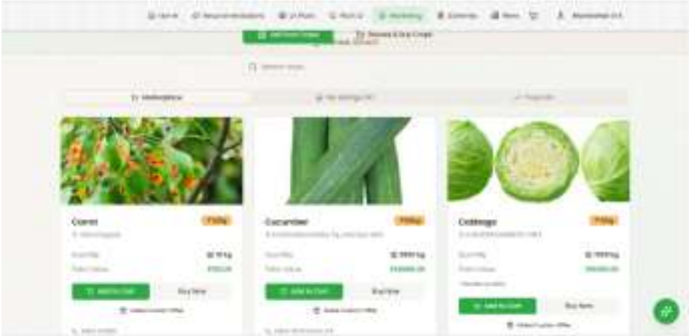
- Dr. Plant

shows the Plant Identification with Plant ID page, which implements the plant species recognition module of the Smart Agriculture System. This feature uses an image-based classifier to identify plants from photos of different organs such as leaves, flowers, fruits, or bark, similar to other AI-based plant ID tools reported in literature.



- Marketing

At the top, a tab labelled "Marketing" is selected, with a search bar that lets users search for specific crops by name. Below this, the page displays individual product cards for each active listing. In the screenshot, three listings are visible: Carrot, Cucumber, and Cabbage. Each card shows a product image, location (village and district), price per kilogram, available quantity, and automatically calculated total value. Two main action buttons—"Add to Cart" and "Buy Now"—enable buyers either to add the item to a shopping cart for later checkout or to proceed directly to purchase. An additional option, “Make Custom Offer”, is provided for negotiating customised quantities or prices, which reflects common practices in agricultural trade.



- Government Schemes

Each scheme card provides a short one-line description, followed by expandable sections for Eligibility Criteria and Benefits & Coverage, enabling farmers to quickly see whether they qualify and what support is offered. A dedicated link “Step-by-Step Guide to Claim” opens a detailed checklist of application steps, and buttons at the bottom show the applicable state (e.g., Maharashtra) and a prominent “Visit Official Portal” button that redirects users to the government website for actual application submission. This design matches recommended practices for digital agriculture portals, where scheme information is simplified and centralised to improve awareness and uptake among farmers.

VI.CONCLUSION

The project successfully designed and implemented an Integrated Full-Stack Smart Agriculture System that brings together AI-based decision support and essential digital services for farmers on a single platform. Using an ANN model for crop recommendation and a CNN model for plant disease detection, the system demonstrates how modern machine learning techniques can be applied to real agricultural problems such as crop planning and early disease diagnosis. The web interface, supported by modules for plant identification, marketplace trading, government schemes navigation, agriculture news, and an AI chatbot, provides a coherent user experience that reduces the fragmentation seen in many existing agricultural applications.
Testing at unit, integration, and system levels showed that the platform meets the specified functional and non-functional requirements, with acceptable response times and correct behaviour across end-to-end workflows. Trial users were able to complete typical tasks—obtaining recommendations, detecting diseases from images, listing products, finding schemes, and interacting with the chatbot—without extensive training, indicating that the interfaces are intuitive and farmer-friendly.
Overall, the project validates the feasibility of building a one-stop smart agriculture portal that combines AI analytics with practical tools and information services. While the current prototype is limited to a defined set of crops, diseases, and regions, it provides a scalable foundation for future enhancements such as Kannada voice interaction, offline-first mobile apps, drone-based crop health monitoring, more detailed fertiliser and pesticide advisory, and integration with IoT sensors. With further data collection, localisation, and field-level evaluation, this system has the potential to evolve into a robust decision-support platform that can significantly improve productivity, reduce risk, and enhance income opportunities for farmers.

VII.REFERENCES

[1] S. Vaishnavi, M. Shobana, R. Sabitha, and S. Karthik, "Agricultural Crop Recommendations based on Productivity and Season," in *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2021, doi: 10.1109/ICACCS51430.2021.9441736.
[2] A. Priyadharshini, A. Kumar, S. Chakraborty, and O. R. Pooniwala, "Intelligent Crop Recommendation System using Machine Learning," in *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, 2021, doi: 10.1109/ICCMC51019.2021.9418375.
[3] A Software Model for Precision Agriculture for Small and Marginal Farmers, Satish Babu
International Centre for Free and Open Source Software (ICFOSS)

[4] Neha Khadke,Divya More, Pooja Patil, Harshali Patil,” Agriculture Decision Support System using Data Mining”,2017

[5] Tanuja R. Patil, 2Shamshuddin K, 3Rajashekhar Patil, 4Sadanand P , “Krushi samriddhi: a decision support system for farmers to get high crop yield”,2016

[6] Anshika Agarwal[1], Y.D.S. Arya[2], Gaurav Agarwal[3], Shruti Agarwal[1] and Kamal Kumar Gola, “A Fuzzy Based Decision Support System for Irrigation Process in Precision Agriculture”,2021

[7] Xiaomin Li, Rihong Zhang,” Integrated multi-dimensional technology of data sensing method in smart agriculture”,2021