AI-Based Smart Farming System

Mouli C, **Dr. Malatesh S H**, **Nadan, Nalin Kumar PS, Ranjith R**

UG Scholar, Head of Department, UG Scholar, UG Scholar , UG Scholar
Computer Science and Engineering,
M S Engineering College, Bengalore, India

ABSTRACT

Agriculture plays a vital role in the economy, yet farmers face challenges such as low crop productivity, unpredictable weather conditions, plant diseases, and fluctuating market prices. Traditional farming methods often lack data-driven decision support, leading to inefficient resource utilization and reduced yields. To address these issues, this project presents an **AI-Driven Farming System using Deep Learning and Machine Learning**.

The system is implemented as a **web-based application using Flask**, with MongoDB for data storage. It provides an easy-to-use interface where users can input soil data, upload crop images, and view recommendations in real time. The proposed solution improves agricultural productivity, reduces crop losses, and supports sustainable farming practices.

This project demonstrates how artificial intelligence can effectively enhance decision-making in agriculture and contribute to smart and sustainable farming solutions.

Keywords: AI, Smart Farming, Crop Recommendation, Disease Detection, Price Prediction, Machine Learning

1. INTRODUCTION

Agriculture is the backbone of the Indian economy and plays a crucial role in ensuring food security and employment. However, traditional farming practices face several challenges such as unpredictable weather conditions, soil degradation, crop diseases, inefficient use of fertilizers, and fluctuating market prices. These challenges often result in reduced crop productivity and financial losses for farmers.

With the rapid advancement of technology, **Artificial Intelligence (AI)** and **Machine Learning (ML)** have emerged as powerful tools to modernize agriculture. By analyzing large volumes of agricultural data, AI can provide accurate insights and recommendations that help farmers make better decisions. Smart farming techniques enable efficient resource utilization, early detection of crop diseases, and improved yield prediction.

This project, titled **"AI-Driven Farming System Using Deep Learning and Machine Learning,"** aims to develop an intelligent software-based solution to assist farmers in crop management. The system uses machine learning algorithms to recommend suitable crops based on soil and weather conditions. Deep learning techniques are employed to detect crop diseases from leaf images at an early stage. Additionally, time-series forecasting is used to predict crop market prices, helping farmers plan sales effectively.

2. METHODOLOGY

The methodology of the proposed project follows a systematic software-based approach to develop an **AI-Driven Smart Farming System**. The system integrates machine learning, deep learning, and web technologies to provide intelligent agricultural decision support.

.

Data Collection

- Soil parameters such as NPK values and moisture levels are collected manually.
- Crop leaf images are collected from standard datasets.
- Weather data is obtained using online APIs.
- Historical crop price data is collected for price prediction.

Data Preprocessing

- Cleaning and normalization of soil and weather data.
- Encoding of categorical values.
- Image preprocessing such as resizing and normalization for disease detection.
- Preparation of time-series data for price prediction.

Model Development

The system uses multiple AI models for different tasks:

- **Random Forest Algorithm**
  Used for crop recommendation based on soil and weather parameters.
- **ResNet9 Deep Learning Model**
  Used for detecting crop diseases from leaf images.
- **SARIMAX Model**
  Used for predicting future crop prices using historical data.

Database Management

- MongoDB is used to store:
  - User details
  - Input data
  - Prediction results
  - Historical records

Frontend Development

- HTML, CSS, and JavaScript are used for UI design.
- User-friendly forms for data input and image upload.
- Results displayed in an understandable format.

Output Generation

The system provides:

- Crop recommendations

- Disease identification and treatment suggestions
- Crop price prediction
- Agricultural insights for farmers

Testing and Validation

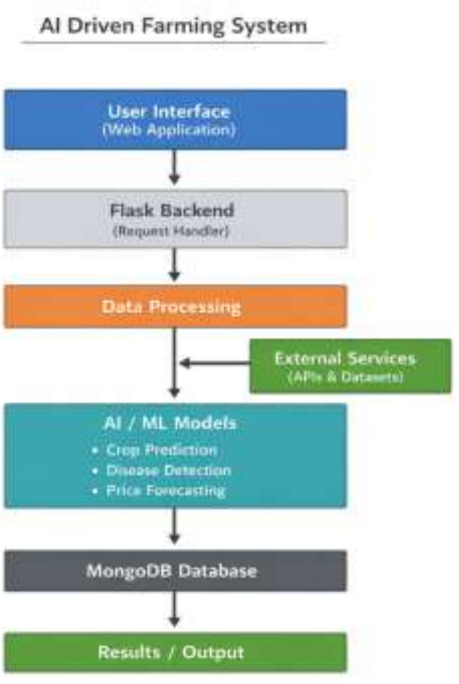
- Unit testing of individual modules
- Integration testing of complete system
- Validation of model accuracy and performance

Deployment

- Application is deployed on a local server or cloud platform.

- Users can access the system through a web browser.

3. Block Diagram



The system works by taking input from the user through a web application. The user enters soil and crop details and uploads crop images. This information is sent to the Flask backend, where the data is processed and prepared for analysis. The processed data is then given to machine learning models to recommend suitable crops, detect plant diseases, and predict crop prices. Weather and market data are also used to improve accuracy. All the information and results are stored in a database. Finally, the results are displayed to the user in a simple and clear format.

Fig.1.System Architecture of AI-Driven Farming System

4. System Architecture

The system architecture defines the interaction between User Input, Flask Backend, Data Processing, AI Models, Database, Output Display.

4.1 User Input

- Web-based application

- User enters soil details and crop information

- User uploads crop image for disease detection

4.2 Flask Backend

- Acts as the main controller

- Receives user input

- Connects frontend with ML models

4.3 Flask Backend

- Cleans and preprocesses input data
- Normalizes soil and weather data
- Preprocesses crop images

4.4 Data Processing

- Cleans and preprocesses input data
- Normalizes soil and weather data
- Preprocesses crop images

4.5 AI Models

- **Random Forest** → Crop recommendation
- **ResNet9** → Disease detection
- **SARIMAX** → Price prediction

4.6 Database

- Stores user details
- Stores prediction results
- Maintains historical data

4.7 Output Display

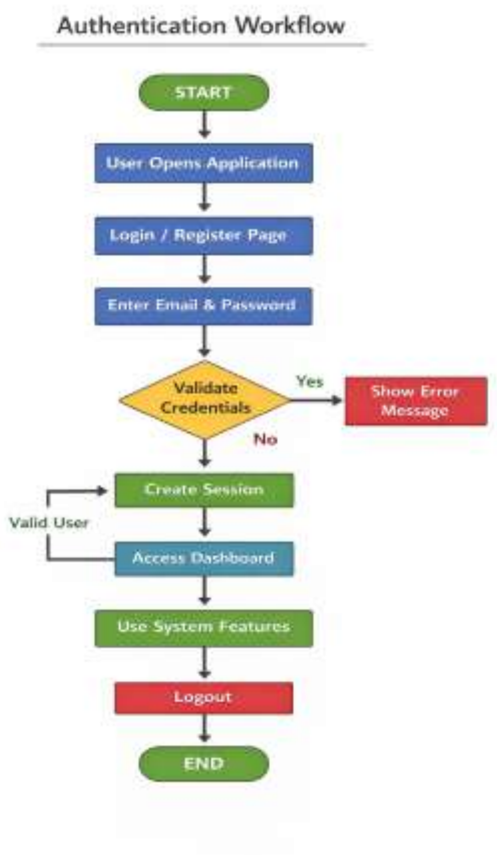
- Displays recommended crops
- Shows disease name and treatment
- Displays predicted market price

5. AUTHENTICATION WORKFLOW

The workflow defines user interaction and system response.

Workflow Steps

1. User opens the application
2. User selects **Login / Register**
3. User enters **Email ID and Password**
4. System validates the entered credentials
5. If credentials are valid → User is authenticated
6. Session is created for the user
7. User is redirected to the dashboard
8. User accesses system features



9. User logs outSession ends

Fig.2. Authentication Flowchart

7. SYSTEM IMPLEMENTATION

The project is implemented as a **web-based application** that integrates machine learning models with a backend server to provide intelligent farming recommendations.

7.1 Technology Used

- Frontend: HTML, CSS, JavaScript
- Backend: Python (Flask Framework)
- Database: MongoDB
- Machine Learning: Random Forest, ResNet9, SARIMAX
- Tools: VS Code, Jupyter Notebook
- latform: Windows

7.2 Implementation Process

Step 1: User Interface Development

- Designed web pages for user login and input
- Forms provided for soil details, crop selection, and image upload
- Results displayed in an easy-to-understand format

Step 2: Backend Development

- Flask framework used to handle user requests
- APIs created to process inputs and return outputs
- Integration of AI models with backend logic

Step 3: Model Integration

- Random Forest model for crop recommendation
- ResNet9 model for disease detection
- SARIMAX model for crop price prediction
- Models loaded and executed during runtime

Step 4: Database Implementation

- MongoDB used to store:
  - User details
  - Input values
  - Prediction results

Step 5: Data Processing

- Cleaning and normalization of data
- Image preprocessing for disease detection
- Formatting of data for ML models

Step 6: Output Generation

- Display of recommended crops
- Disease name and treatment
- Predicted crop price

7.3 System Execution Flow

- User logs in to the system
- Inputs are provided through the web interface
- Backend processes the data
- ML models generate predictions
- Results are stored in the database
- Output is displayed to the user

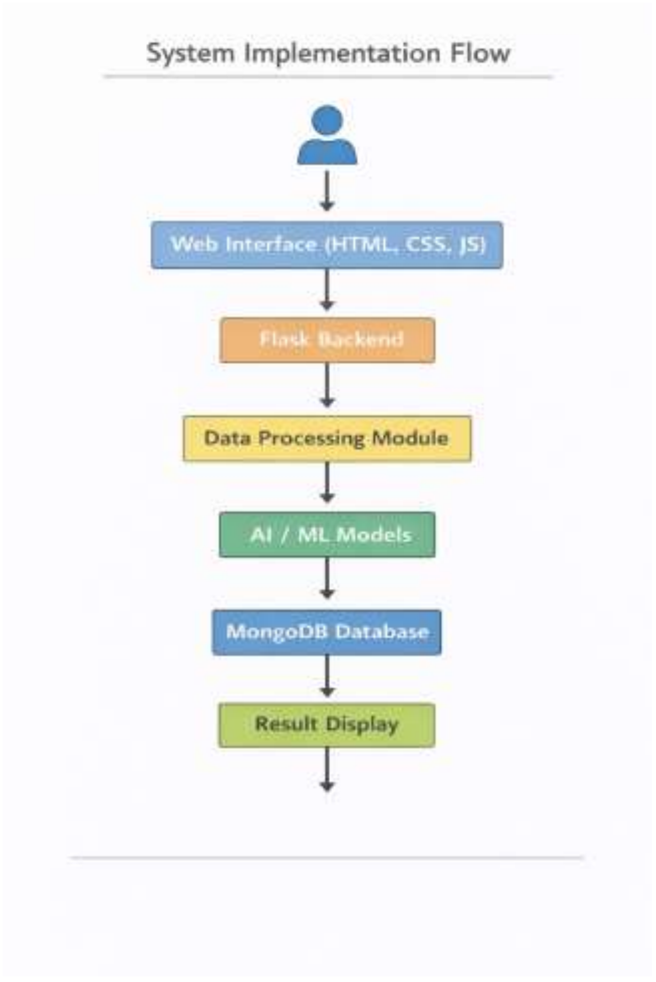


Fig.3. System Implementation Flow Diagram of AI-Driven Farming System

8. Results

The proposed **AI-Driven Farming System** was successfully designed and implemented as a software-based solution. The system was tested with different inputs and datasets, and the following results were obtained:

The **crop recommendation module** accurately suggested suitable crops based on soil parameters and weather conditions using the Random Forest algorithm.

The **disease detection module** successfully identified crop diseases from uploaded leaf images using the ResNet9 deep learning model and provided appropriate treatment suggestions.

The **price prediction module** effectively forecasted future crop prices using the SARIMAX model, helping in better market planning.

The **web application interface** allowed users to easily enter inputs, upload images, and view results in real time.

The **MongoDB database** stored user inputs and prediction results efficiently, enabling data retrieval and history tracking.

Overall, the system demonstrated **accurate predictions, fast response time, and user-friendly interaction**, proving the effectiveness of AI and machine learning in smart farming applications.

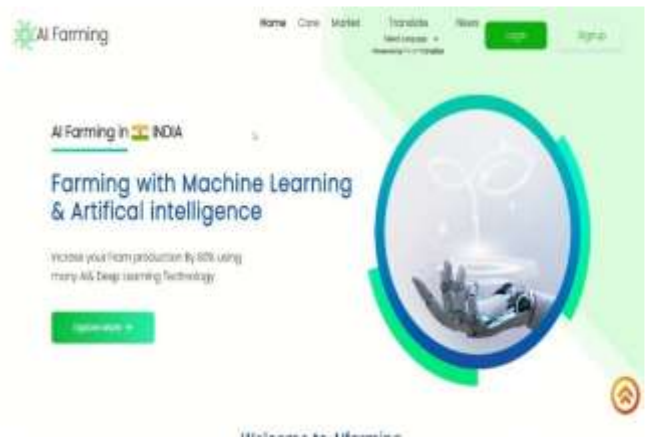


Fig.4. Home Page of AI-Driven Smart Farming Web Application

9. Future Scope

The proposed AI-Driven Farming System can be further enhanced and expanded in several ways to improve accuracy, usability, and real-world impact.

- Integration of IoT Sensors
- Mobile Application Development
- Advanced AI Models
- Multi-Language Support
- Real-Time Market Integration
- Automated Irrigation System
- Government Scheme Integration
- Scalability and Cloud Deployment

10. Conclusion

The AI-Driven Farming System developed in this project provides an effective software-based solution to support farmers in making data-driven agricultural decisions. By integrating machine learning and deep learning techniques, the system successfully recommends suitable crops, detects plant diseases from leaf images, and predicts crop market prices through a user-friendly web application. The use of Random Forest for crop recommendation, ResNet9 for disease detection, and SARIMAX for price forecasting demonstrates the practical application of AI in agriculture. The Flask-based backend and MongoDB database ensure smooth processing, efficient data storage, and quick response time. Overall, the project highlights how artificial intelligence can enhance agricultural productivity, reduce losses, and promote smart and sustainable farming practices.

11. References

1. Tom M. Mitchell, *Machine Learning*, McGraw-Hill Education, 1997.
2. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press, 2016.
3. Breiman, L., "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
4. He, K., Zhang, X., Ren, S., and Sun, J., "Deep Residual Learning for Image Recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
5. Box, G. E. P., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M., *Time Series Analysis: Forecasting and Control*, Wiley, 2015.
6. Dr. Malatesh S. H., Mohanty, S. P., Hughes, D. P., and Salathé, M., "Using Deep Learning for Image-Based Plant Disease Detection," *Frontiers in Plant Science*, 2016.
7. Flask Documentation – https://flask.palletsprojects.com/

8. MongoDB Documentation – https://www.mongodb.com/docs/
9. Kaggle Datasets – Crop Recommendation and Plant Disease Datasets, https://www.kaggle.com/
10. OpenWeather API Documentation – https://openweathermap.org/api