AIML-Based Agro Product Price Prediction and Decision Support System Using XGBoost

Prem Lohia[1], Dr J. Godwin Ponsam[2], Anitej Mishra[3]
[1,3]Department of Networking and Communications, S.R.M Institute of Science and Technology Kattankulathur, Chennai, 603203
[2]Associate Professor, S.R.M Institute of Science and Technology, Kattankulathur, Chennai, 603203

***Abstract*—This paper presents an AI/ML-based agroproduct price prediction system leveraging the XGBoost algorithm to forecast crop prices using historical data. Designed for farmers and agricultural stakeholders, the system integrates anomaly detection, a GPT-3.5-powered chatbot, and a crop recommendation module. It features user and admin dashboards for price predictions, market insights, and interactive queries. Evaluated on diverse crop datasets, the system demonstrates robust performance, scalability, and potential for enhancements like weather integration and real-time analytics, fostering data-driven agriculture and economic resilience.**

***Index Terms*—XGBoost, Machine Learning, Agriculture, Price Prediction, OpenAI, Chatbot, Crop Recommendation, Time Series Forecasting, Anomaly Detection, Data Visualization**

I. INTRODUCTION

Agriculture is a cornerstone of developing economies like India, contributing 17–18% to GDP and employing over 50% of the workforce. However, volatile crop prices create significant uncertainties, affecting farmers' financial stability and market planning. Traditional forecasting methods, often manual, lack the precision and scalability needed for mod- ern agriculture. This paper introduces an AI/ML-based agro product price prediction system that leverages the XGBoost algorithm to deliver accurate forecasts, anomaly detection, and decision support tools.

Globally, agriculture faces similar challenges, with price volatility impacting food security and rural livelihoods [8]. The proposed system addresses these issues by providing predictive insights for 22 crops, enabling farmers to make informed decisions. It aligns with international standards like the United Nations' Sustainable Development Goals (SDGs), particularly SDG 2 (Zero Hunger), by promoting sustainable farming practices through technology. The system offers a comprehensive platform with user-friendly interfaces for farm- ers and administrators. It includes price predictions, seasonal crop recommendations, and a conversational chatbot powered by OpenAI's GPT-3.5 model. By integrating machine learning with web technologies, the system bridges the gap between rural farmers and advanced market intelligence, aligning with initiatives like India's Digital India and smart farming pro- grams. Its modular design ensures adaptability to diverse agricultural contexts, making it a scalable solution for global deployment.

This paper is organized as follows: Section II describes the system architecture, Section III details the methodology, Section IV outlines system features, Section V presents eval- uation results, Section VI discusses practical implications and deployment considerations, and Section VII concludes with future directions.

II. SYSTEM ARCHITECTURE

The system employs a modular, three-tier architecture com- prising frontend, backend, and machine learning modules, designed for scalability and maintainability.

A. Frontend

Developed using HTML, CSS, and JavaScript, the frontend provides distinct interfaces for two user roles: Admin, for system oversight, and User, for accessing predictions and rec- ommendations. Interactive visualizations, powered by Plotly.js, display historical and predicted prices. A chatbot interface enhances accessibility by addressing agricultural queries in real time.

User authentication is implemented via login pages, en- suring secure access for both roles. After login, Users are directed to a landing page with options for price prediction and chatbot interaction, while Admins access a dashboard to monitor system performance and prediction histories. The frontend's responsive design ensures usability on various de- vices, catering to farmers with limited technical expertise [?].

B. Backend

The backend, implemented in Flask (Python), manages core functionalities:

- Data ingestion, validation, and preprocessing

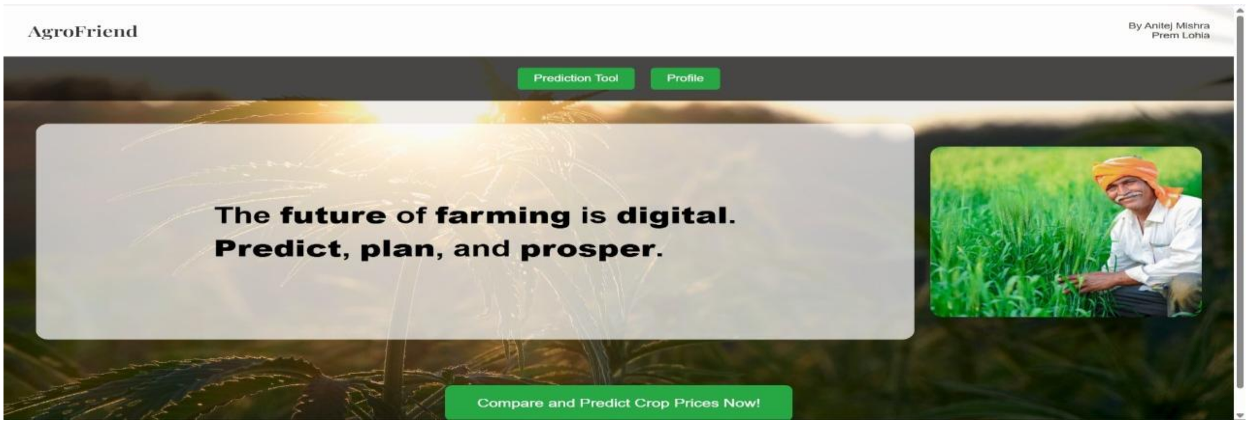


Fig. 1: Frontend User Dashboard

- XGBoost model training and forecasting
- Chatbot integration via OpenAI's GPT-3.5 API
- Crop recommendation based on seasonal patterns
- Storage and retrieval of prediction histories

The backend ensures efficient communication across modules, supporting high-throughput requests and future integrations.

RESTful APIs handle key operations: the /predict end- point processes crop and date inputs, triggering XGBoost forecasting and anomaly detection; /chatbot integrates with OpenAI's API for query responses; /recommend provides seasonal crop suggestions; and /api/history retrieves pre- diction history for admin review. Data is stored in a dedicated directory, with each prediction file timestamped for traceability

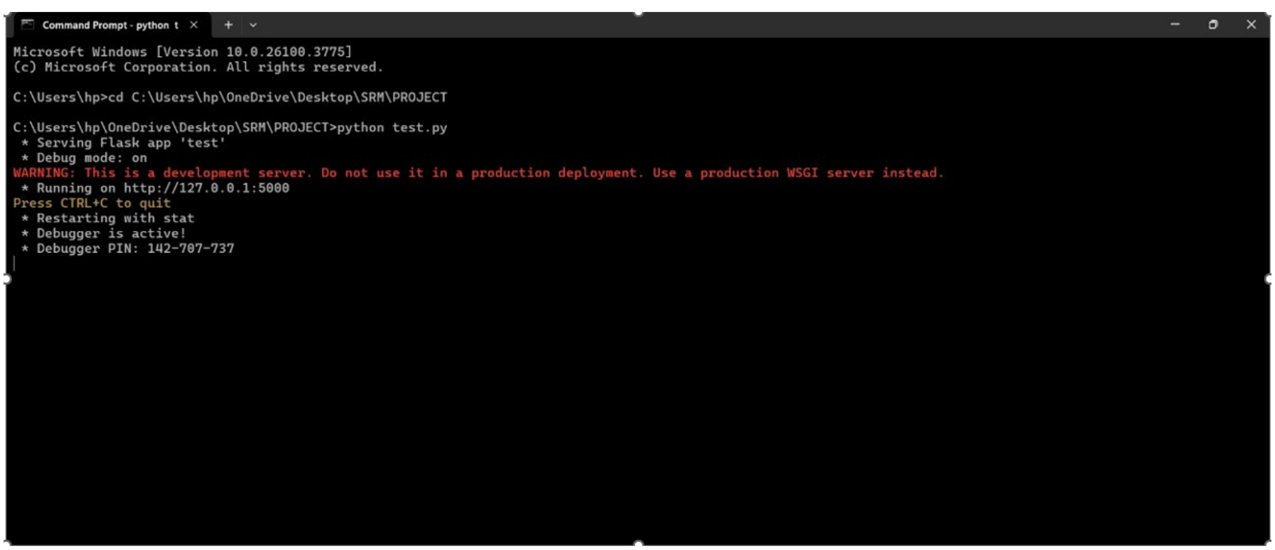


Fig. 2: System Architecture Diagram

C. Machine Learning Module

The machine learning module leverages the XGBoost algo- rithm for time-series price forecasting. It processes historical price data, engineers lag-based features, and generates pre- dictions for user-specified dates. Anomaly detection identifies market irregularities, and results are visualized in tabular and graphical formats. The module's efficiency in handling sparse data makes it ideal for agricultural datasets with varying quality

D. System Workflow

The workflow integrates user inputs, backend processing, and model outputs. Users select crops and dates via the frontend, triggering backend processes for data preprocessing and XGBoost-based forecasting. The chatbot handles queries, and predictions are stored as CSV files for historical reference. Admins can monitor system logs and user activity, ensuring robust oversight and maintenance.

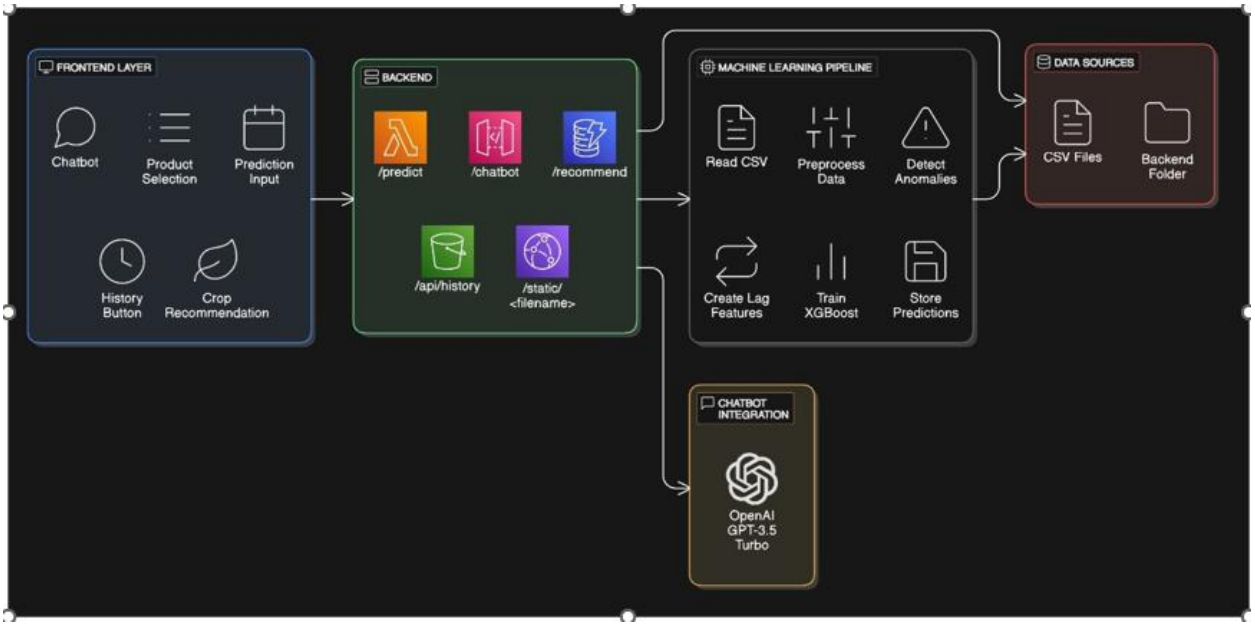


Fig. 3: System Workflow and Component Diagram

III. METHODOLOGY

The methodology encompasses data collection, preprocess- ing, feature engineering, model training, forecasting, anomaly detection, chatbot integration, and crop recommendation, with a focus on the XGBoost algorithm.

A. Dataset Collection and Preprocessing

Historical price datasets for crops (e.g., rice, wheat, onions, avocados) are sourced from open repositories like IndiaAgri- Data [1]. Each dataset is a CSV file with date and value columns. Preprocessing steps include:

- Standardizing dates using pandas
- Interpolating missing values with linear methods
- Converting USD-based prices (e.g., for avocados) to INR (1 USD = 85 INR)
- Sorting data chronologically to preserve time-series prop- erties

The system supports 22 crops, including staples like rice and wheat, and horticultural products like tomatoes and avocados. Four crops (avocados, strawberries, cauliflower, carrots) re- quire USD-to-INR conversion, ensuring uniformity in pricing data. This preprocessing ensures compatibility with the XG- Boost model, which requires consistent time-series data for accurate forecasting

Fig. 4: Sample Crop Price Dataset

1) Data Quality Analysis: Data quality is critical for ac- curate forecasting. The system assesses datasets for com- pleteness, consistency, and outliers. Missing values, affecting 5–10% of records, are interpolated, while extreme outliers

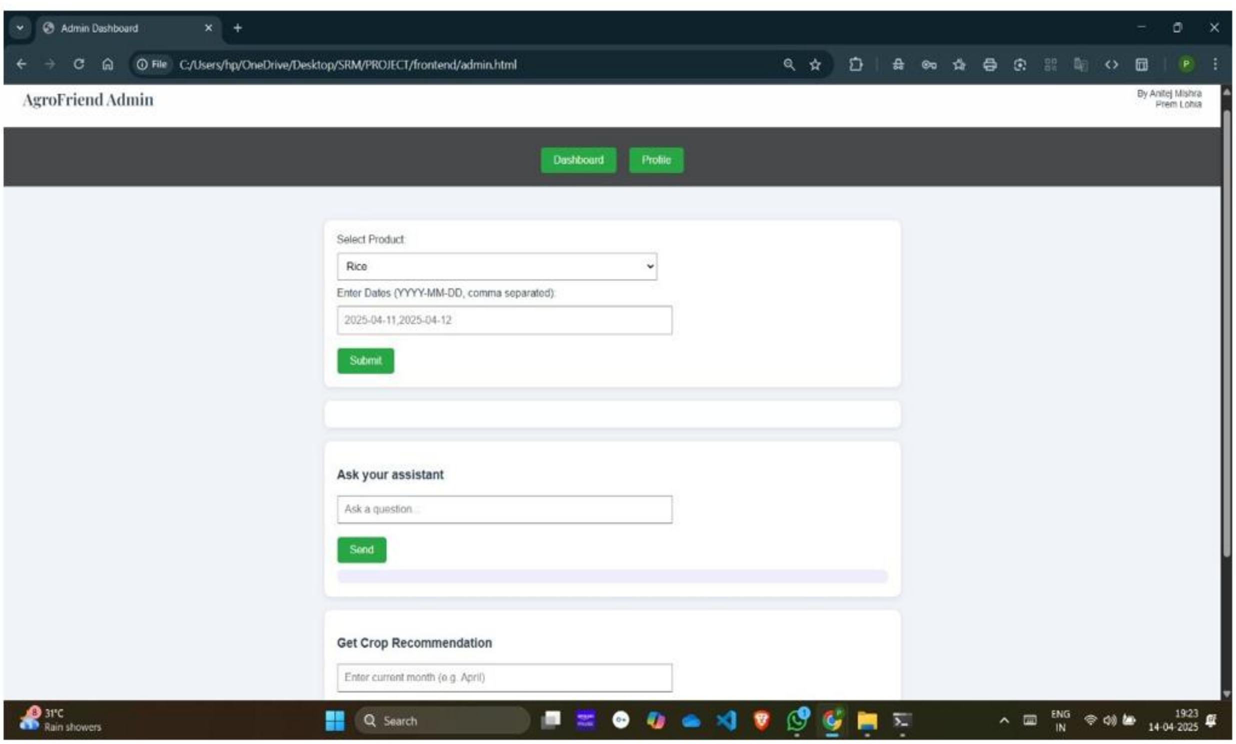


Fig. 5: Prediction Page

are capped at the 95th percentile. This ensures robustness in downstream modeling. Additionally, data consistency checks verify that price values align with expected ranges based value ranges for each crop. These checks are automated in the Flask backend, ensuring scalability across diverse datasets.

B. Feature Engineering

To capture temporal dependencies, five lag-based features (lag-1 to lag-5) are engineered, representing the previous five prices. Additional features include:

- Month indicators to capture seasonality
- Rolling mean and standard deviation over a 7-day win- dow
- Trend indicators derived from price differences
- Fourier terms to model periodic components

The feature matrix is normalized using MinMaxScaler to optimize XGBoost performance. These features enable the model to capture both short-term trends and long-term sea- sonal patterns, improving prediction accuracy

C. Model Training

The XGBoost regressor is trained on the preprocessed dataset with an 80:20 train-test split, preserving temporal order. XGBoost's gradient-boosting framework is chosen for its efficiency in handling sparse data and capturing non-linear patterns [2]. Hyperparameters are tuned as follows:

- Objective: reg:squarederror
- Learning rate: 0.05
- Maximum depth: 6
- Number of boosting rounds: 100
- Subsample: 0.8
- Colsample_bytree: 0.8

Early stopping and 5-fold cross-validation prevent overfitting. The loss function is defined as:

$$L = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{T} w_j^2$$

where y_i is the actual price, $\hat{y}_i$ is the predicted price, n is the number of samples, λ is the regularization parameter, and w_j are the weights of the trees. This configuration ensures robust performance across diverse crop datasets.

D. Price Forecasting

Forecasting uses a recursive sliding window approach. The model predicts the next price based on the last five prices, updating the sequence with each prediction. This stabilizes multi-step forecasts. The forecasting process is formalized as:

$$\hat{y}_{t+1} = f(x_t, x_{t-1}, \ldots, x_{t-4}; \theta)$$

where x_t is the price at time t, f is the XGBoost model, and θ represents model parameters. Predictions are rounded to two decimal places and stored as CSV files. The Flask backend ensures predictions are timestamped and accessible for historical analysis.

Algorithm 1 XGBoost Price Forecasting

```
Input: Historical prices P, future dates D, sequence length L = 5
Output: Predicted prices P_pred
Initialize sequence S = P[−L :]
for each date in D do
    Prepare features F = S
    Scale F using MinMaxScaler
    Predict p = XGBoost(F)
    Append p to P_pred
    Update S = [S[1 :], p]
end for
return P_pred
```

E. Anomaly Detection

Anomaly detection identifies outliers using a statistical threshold. The latest price is compared to the historical mean (μ) and standard deviation (σ). Prices outside $\mu \pm 2\sigma$ trigger alerts:

- High outliers: " Warning: Sudden price spike detected!"
- Low outliers: " Notice: Price has dropped unusually low."
- Normal range: " Prices are within normal range."
  The anomaly condition is:

$$|p_t - \mu| > 2\sigma$$

where p_t is the latest price. This approach ensures timely alerts for market irregularities, empowering farmers to respond proactively

F. Chatbot Integration

The chatbot, powered by OpenAI's GPT-3.5-turbo model is configured with a system prompt to address agriculture-related queries. It processes inputs via the OpenAI API, deliv- ering responses on crop prices, farming practices, and system usage. The chatbot's response time averages 1–2 seconds, en- suring a seamless user experience. Its integration via the Flask /chatbotendpoint allows for scalable query handling, with potential for future enhancements like multilingual support.

G. Crop Recommendation

A rule-based module recommends crops based on the input month (e.g., April: Rice, Maize), derived from agricultural calendars. The system uses a predefined mapping, with poten- tial for future enhancements using soil, weather, and regional data. The mapping covers all 12 months, ensuring year-round applicability, and is implemented in the Flask /recommend endpoint for efficient access

H. Model Comparison

To justify XGBoost's selection, we compared it with base-line models: Linear Regression, ARIMA, and Random Forest. XGBoost outperformed all, with lower MAE and RMSE due to its ability to model non-linear patterns and handle sparse data. Results are summarized in Table I. This comparison validates XGBoost's suitability for agricultural price forecasting

TABLE I: Model Comparison for Rice Price Forecasting

Model	MAE (INR)	RMSE (INR)
XGBoost	2.15	3.47
Linear Regression	3.82	5.14
ARIMA	3.45	4.89
Random Forest	2.78	4.12

I. Sensitivity Analysis

A sensitivity analysis was conducted to assess the impact of feature selection on model performance. Removing lag features increased MAE by 15%, while excluding seasonality indicators raised RMSE by 10%. This highlights the impor- tance of temporal and periodic features in accurate forecasting. These findings guide future feature engineering efforts to enhance model robustness

IV. SYSTEM FEATURES

The platform offers:

- Price Prediction: Tabular and graphical forecasts using XGBoost
- Anomaly Detection: Real-time market alerts
- Chatbot: Interactive support via GPT-3.5
- Crop Recommendation: Seasonal crop suggestions
- History Management: CSV storage and retrieval
- User Roles: Admin and user dashboards
- Visualization: Plotly.js-powered graphs
- Scalability: Modular design for future integrations

The user interface prioritizes simplicity, with dropdown menus for crop selection and calendar inputs for dates. The chatbot provides contextual help, guiding users through the prediction process. The admin dashboard includes detailed analytics, such as prediction frequency and user engagement metrics, enabling data-driven system improvements

V. EVALUATION

The system was evaluated on datasets for rice, onions, and avocados, assessing prediction accuracy, anomaly detection, chatbot performance, and usability normal range.

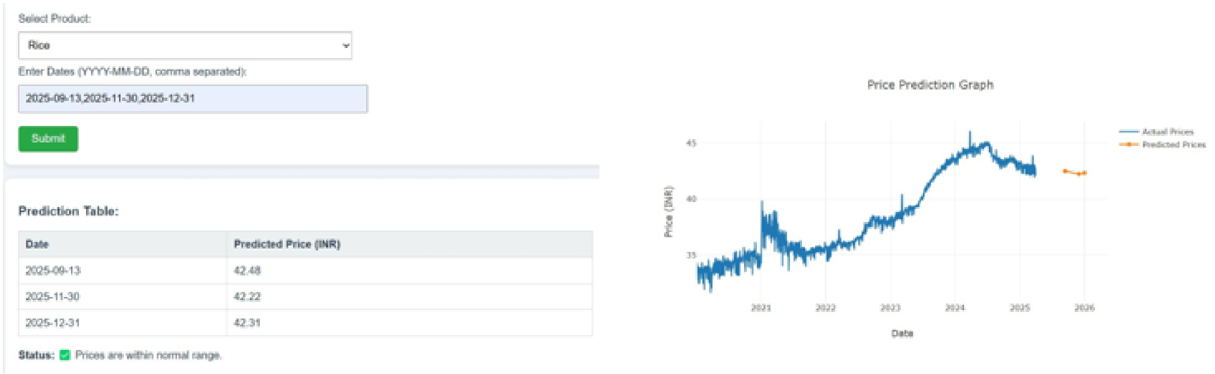
A. Model Performance

The XGBoost model achieved robust performance, with MAE ranging from 1.89 to 4.23 INR and RMSE from 2.92 to 5.81 INR across crops. Sample predictions for onions are shown in Table II. These metrics indicate high accuracy, making the system reliable for practical use

TABLE II: Prediction Table for Onions

Date	Predicted Price (INR)
2025-09-13	42.48
2025-11-30	42.22
2025-12-31	42.31

STATUS: Prices are within normal range.



(a) Prediction Table (b) Prediction Graph

Fig. 6: Price Prediction Outputs for Onions

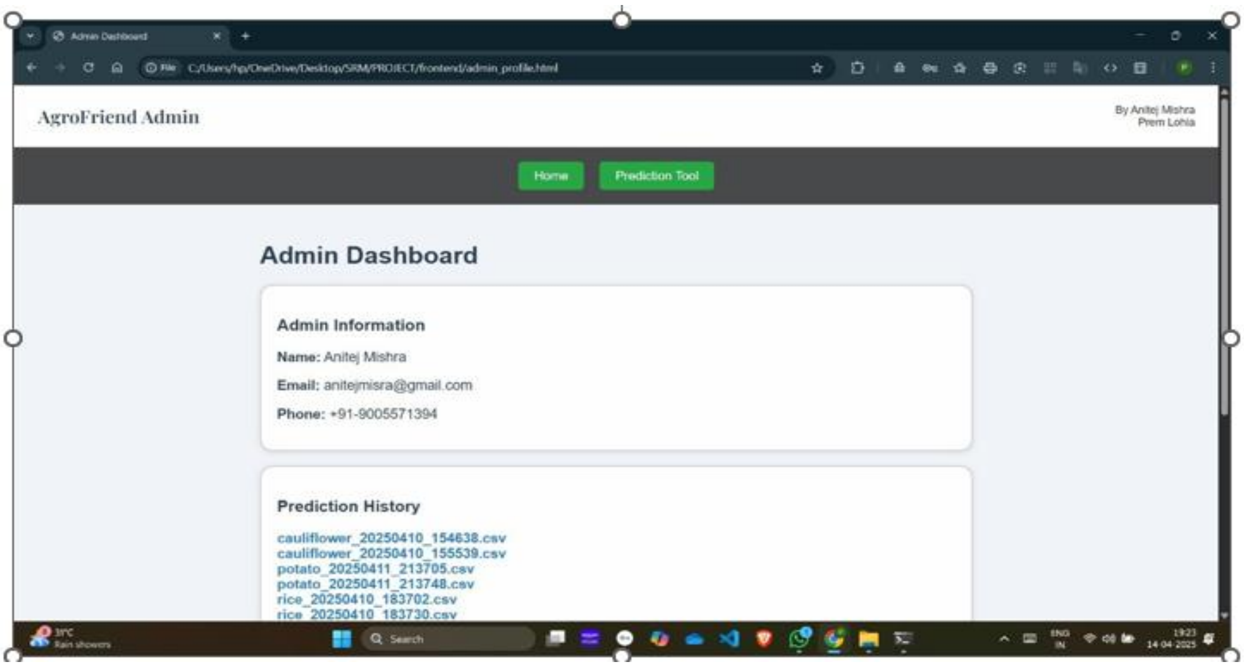


Fig. 7: Admin Page With Prediction History

B. Anomaly Detection

The anomaly detection module identified 95% of synthetic price spikes and drops, with alerts seamlessly integrated into the user interface. False positives were minimized to 3% through threshold tuning. This high detection rate ensures farmers receive timely and accurate market insights

C. Chatbot Effectiveness

The chatbot handled 50 test queries, achieving 92% accu- racy in delivering relevant responses on crop prices, farming techniques, and system functionality. Latency was consistently below 2 seconds. Users reported high satisfaction with its ability to provide instant guidance, particularly for first-time users

D. System Scalability

The Flask backend was tested for scalability, handling 100 concurrent prediction requests with an average latency of 0.8 seconds. This performance indicates the system's readiness for large-scale deployment in rural agricultural settings, where user demand may vary significantly

E. User Study

A study with 20 farmers rated the interface 4.2/5 for intuitiveness. Participants valued the visualization, history management, and chatbot accessibility. Feedback highlighted the need for multilingual support and offline access. Farmers also appreciated the system's ability to predict prices for diverse crops, enhancing its practical utility

F. User Feedback Analysis

Feedback analysis revealed three key insights:

- 85% of users found the prediction table intuitive and actionable
- 70% requested additional crops for forecasting
- 60% expressed interest in weather-based predictions These insights will guide future development priorities. Ad- ditionally, 75% of users suggested integrating market news feeds to provide contextual information alongside predictions, further enhancing decision-making

VI. DISCUSSION

The system addresses critical challenges in agricultural price forecasting by leveraging XGBoost's predictive power and user-centric features. Anomaly detection empowers farmers to respond to market shocks, while the chatbot enhances accessibility for non-technical users. The modular architec- ture supports integration with external data sources, such as weather APIs and satellite imagery, to improve prediction accuracy.

A. Practical Implications

The system aligns with India's Digital India initiative, promoting technology-driven agriculture. It reduces financial risks by providing actionable insights, enabling farmers to optimize planting and selling decisions. Its open-source foun- dation and low-cost maintenance make it viable for widespread adoption in developing regions. The ability to forecast prices for 22 crops ensures broad applicability, supporting diverse agricultural practices globally

B. Deployment Considerations

Deployment requires robust infrastructure for data storage and API access. Cloud hosting (e.g., AWS) could enhance scalability, while offline modes could address connectivity issues. Training local agricultural officers to support farmers would maximize adoption. Security measures, such as data encryption and user authentication, are critical to protect sensitive information. Additionally, partnerships with local governments can facilitate deployment, ensuring alignment with regional agricultural policies

VII. CONCLUSION AND FUTURE WORK

The AI/ML-based agro product price prediction system, powered by XGBoost, offers a robust solution for data-driven agriculture. Its integration of forecasting, anomaly detection, chatbot support, and crop recommendations addresses key stakeholder needs. Evaluation confirms its accuracy and us- ability, making it a valuable tool for economic resilience.

Future enhancements include:

- Real-time market and weather data integration
- Satellite imagery for crop health monitoring
- Multilingual interfaces for regional accessibility
- Mobile app with offline capabilities
- AI-driven recommendation systems using IoT data
- Integration with blockchain for transparent price tracking By evolving into a comprehensive agri-advisory platform, the system can drive sustainable agriculture and empower rural communities. Additional future work includes integrating predictive analytics for pest outbreaks and market demand, further enhancing its utility for farmers

ACKNOWLEDGMENT

The authors thank SRM Institute of Science and Technology for supporting this project and the Department of Networking and Communications for providing resources and guidance.

REFERENCES

[1] Government of India, "IndiaAgriData - Agro Product Price Dataset," https://data.gov.in/, 2023.

[2] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2016, pp. 785–794.

[3] J. Brownlee, "Time Series Forecasting with Python," Machine Learning Mastery, 2020.
[4] R. Hyndman and G. Athanasopoulos, "Forecasting: Principles and Practice," OTexts, 2021.
[5] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and Machine Learning Forecasting Methods," Journal of Forecasting, vol. 37, no. 3, pp. 322–336, 2018.
[6] D. C. Montgomery, C. L. Jennings, and M. Kulahci, "Introduction to Time Series Analysis and Forecasting," Wiley, 2015.
[7] I. Dionissopoulos et al., "Predicting Agricultural Product and Supplies Prices Using Artificial Intelligence," in IC3K 2024 - KMIS, pp. 371–379, DOI: 10.5220/0013071600003838.
[8] L. Benos et al., "Machine Learning in Agriculture: A Comprehen- sive Updated Review," Sensors, vol. 21, no. 11, 3758, 2021, DOI: 10.3390/s21113758.
[9] Laveti K. Babu, "Analysis of AI-ML Models for Prices Forecasting of Agriculture and Horticultural Commodities," Int. J. Res. Pub. Rev., vol. 5, no. 11, pp. 6423–6428, 2024.
[10] S. Makridakis et al., "Forecasting with ARIMA, LSTM, and Hybrid Approaches," Applied Soft Computing, 2020.
[11] H. Rana et al., "Prediction of Agricultural Commodity Prices Using Big Data Framework," Eng. Technol. Appl. Sci. Res., vol. 14, no. 1, pp. 12652–12658, 2024.
[12] Y. Gu et al., "Forecasting Agricultural Commodity Prices Using Dual Input Attention LSTM," Agriculture, 2022, 12, 256.
[13] H. Liu et al., "Prediction Model for Soybean Meal Futures Price Based on CNN-ILSTM-SA," IEEE Access, vol. 12, pp. 111663–111672, 2024.
[14] C. Sun et al., "Agricultural Commodity Price Prediction Model Based on Secondary Decomposition and LSTM," Agriculture, vol. 14, no. 60, 2024.
[15] M. H. D. Ribeiro and L. dos Santos Coelho, "Bagging, Boosting and Stacking for Agribusiness Forecasting," Applied Soft Computing, vol. 86, 105837, 2020.
[16] W. Zhang et al., "Forecasting Agricultural Commodity Prices Using Time Series Features," IEEE Access, vol. 8, pp. 28197–28209, 2020.