International Journal on Advanced Computer Theory and Engineering
ISSN: 2319 - 2526
Volume 15 Issue 2s (2026) | Pages: 135-144

HelloFarmer: An Intelligent Web-Based Agricultural Decision Support Platform for Climate Prediction, Crop Insights, and Market Price Forecasting

Sidhanth Tyagi[1], Aryan Landge[2], Adwait Sable[3], Priya Vatsala[4]

[1,2,3] Student Department of Artificial Intelligence and Data Science, GSMCOE Balewadi, Maharashtra, India
[4]Assistant Professor, Department of Artificial Intelligence and Data Science, GSMCOE Balewadi, Maharashtra, India
Email: sidtyogi@gmail.com, aryzusl@gmail.com, adwaitas04@gmail.com, vatsalapriya@gmail.com

Peer Review Information

Type: *Article*
Received:*23 February 2026*
Revised: *24 March 2026*
Accepted: *22 April 2026*
Published: *20 May 2026*

Abstract

Agriculture remains one of the most critical yet technologically underserved sectors globally, accounting for the livelihoods of over 1.3 billion people. Farmers face mounting challenges from unpredictable climate patterns, volatile commodity markets, lack of timely crop-specific guidance, and limited access to data-driven tools. Traditional approaches to obtaining weather forecasts, crop advisories, and price information are fragmented, time-consuming, and often inaccessible to smallholder farmers in rural regions. This paper presents HelloFarmer, a comprehensive web-based agricultural decision support platform that integrates three core intelligence modules into a unified, accessible interface: (1) a climate prediction engine that forecasts temperature, rainfall, and humidity using LSTM-based deep learning and Random Forest models trained on historical meteorological data; (2) a crop insight engine that generates personalized crop recommendations and growing advisories by combining real-time climate forecasts with soil parameter inputs using weighted multi-feature scoring; and (3) a market price forecasting module that leverages XGBoost and Facebook Prophet models trained on historical mandi price records to predict near-term commodity prices with quantified confidence intervals. HelloFarmer is built on a Vue.js 3 frontend, a FastAPI asynchronous backend, and a PostgreSQL database, ensuring responsiveness and scalability. Experimental evaluation on Maharashtra regional datasets demonstrates climate forecast MAE below 2°C, crop recommendation top-3 accuracy of 89.2%, and price prediction R^2 scores exceeding 0.86 across multiple commodity-market pairs. User evaluation with 30 farmers and agricultural students confirmed high task completion rates without prior training, validating the platform's core objective of democratizing agricultural intelligence for non-expert users.

Keywords: Agricultural Decision Support; Climate Prediction; Crop Recommendation; Price Forecasting; LSTM; XGBoost; Prophet; FastAPI; Vue.js; Machine Learning; Precision Agriculture; AutoML

How to Cite This Article

Tyagi, S., Landge, A., Sable, A., & Vatsala, P. (2026). *HelloFarmer: An Intelligent Web-Based Agricultural Decision Support Platform for Climate Prediction, Crop Insights, and Market Price Forecasting. International Journal on Advanced Computer Theory and Engineering*, 15(2s), 135–144.

Introduction

Agriculture is the backbone of developing economies, yet it remains one of the sectors least served by modern information technology. In India alone, farming employs approximately 58% of the rural workforce and contributes nearly 18% of GDP [6]. Despite this significance, the vast majority of smallholder farmers make planting, irrigation, and selling decisions based on intuition, local tradition, or word-of-mouth, with little access to scientifically grounded, real-time data. The consequences are severe: crop failures from untimely decisions, post-harvest losses from inadequate price awareness, and persistent income volatility that traps farm families in cycles of debt.

The three most critical information needs that farmers consistently identify are: (i) reliable short-term weather forecasts to guide sowing and irrigation; (ii) guidance on which crops are best suited to current and upcoming conditions; and (iii) market price forecasts to determine when and where to sell. Currently, these needs are served by disparate, siloed sources—national meteorological departments for weather, government extension services for crop guidance, and unverified trader networks for price information. None of these channels are integrated, real-time, or accessible through a single interface.

The rapid advancement of machine learning, time-series forecasting, and web application frameworks now makes it technically feasible to build a unified, intelligent agricultural platform that addresses all three needs simultaneously. This paper presents HelloFarmer, a browser-based agricultural decision support system that combines climate prediction, crop insights, and market price forecasting into a single, zero-installation platform. HelloFarmer is designed for non-expert users—farmers, cooperative societies, agricultural students, and extension workers—and requires no programming knowledge or specialized data science background.

The primary contributions of this paper are: (1) the design and implementation of a three-module integrated agricultural intelligence architecture; (2) an LSTM and ensemble learning pipeline for district-level weather prediction tailored to agricultural use cases; (3) a multi-feature crop scoring engine that combines climate forecasts with soil inputs for context-aware recommendations; (4) a commodity price forecasting framework combining gradient-boosted trees with seasonal decomposition models; and (5) a user study validating the platform’s accessibility and effectiveness with real farming community participants.

The remainder of this paper is structured as follows. Section II surveys related work in agricultural decision support, climate modeling, and price forecasting. Section III describes the methodology underpinning each prediction module. Section IV details the system architecture. Section V covers testing and validation. Section VI presents experimental results. Section VII discusses research gaps and future directions, and Section VIII concludes.

Literature Survey

Agricultural decision support has been an active area of research for over two decades, yet the transition from academic prototypes to accessible, production-ready platforms for smallholder farmers remains incomplete. This section reviews the state of the art across the three core domains of HelloFarmer: climate prediction, crop recommendation, and price forecasting.

Climate Prediction for Agriculture

Numerical weather prediction (NWP) models operated by national agencies such as the India Meteorological Department (IMD) provide coarse-grid forecasts that are accurate at a regional scale but insufficient for field-level agricultural decisions [4]. The resolution gap—often 25 to 50 km grid spacing—means that microclimatic variations critical to crop growth are entirely missed.

Recent machine learning approaches have demonstrated promising results in downscaling and post-processing NWP outputs. LSTM-based recurrent neural networks have been shown to outperform ARIMA and classical regression models for temperature and rainfall prediction at station level, especially for multi-step ahead forecasts of 3 to 14 days [12]. Hybrid models that combine LSTM with attention mechanisms or ensemble methods have further improved forecast skill for monsoon-dominant regions. However, these models are typically published as isolated research artifacts, not integrated into user-facing agricultural platforms.

Crop Recommendation Systems

Early crop recommendation systems relied on rule-based expert systems encoding agronomy domain knowledge. Contemporary approaches use supervised classifiers—decision trees, random forests, naive Bayes—trained on soil and climate datasets to predict optimal crops [1]. Datasets such as the Crop Recommendation Dataset (soil NPK, temperature, humidity, pH, rainfall) have enabled models achieving classification accuracies above 95% in controlled benchmark settings.

Commercial platforms such as Cropin and IFFCO Kisan offer crop advisory features but require internet connectivity and farmer registration, with advisory content often generated centrally without personalization to individual soil or microclimatic conditions. Platforms such as AgriSuite and FarmStack provide richer feature sets but target agribusiness enterprises, with pricing structures prohibitive for individual smallholder farmers [7]. None of these platforms integrate dynamic, real-time climate forecasts into their crop recommendation pipelines.

Agricultural Price Forecasting

Commodity price forecasting for agricultural markets has been studied extensively using time-series methods. Classical approaches include ARIMA, SARIMA, and exponential smoothing, which capture autocorrelation and seasonality in price series [3]. These methods perform adequately for stable, low-volatility commodities but degrade rapidly for perishable crops such as tomatoes, onions, and leafy vegetables, where prices exhibit high intra-season volatility driven by supply shocks.

Machine learning approaches—gradient boosted trees, support vector regression, and neural networks—have demonstrated improved performance by incorporating exogenous features such as arrival quantities, rainfall, fuel costs, and lagged prices across interconnected markets. The Facebook Prophet model, designed for time series with strong seasonal effects and irregular holidays, has shown practical effectiveness for agricultural price series in developing countries. However, these models are rarely packaged into accessible, production-deployed platforms integrated with real mandi data feeds.

Gaps Addressed by HelloFarmer

A review of existing platforms reveals four persistent gaps: (1) integration—no freely available platform combines climate prediction, crop recommendation, and price forecasting; (2) accessibility—platforms that do offer multiple features require technical expertise or paid subscriptions; (3) real-time personalization—recommendations are generic rather than derived from the user's specific location and season; and (4) transparency—black-box outputs without confidence intervals or explanatory context reduce farmer trust. HelloFarmer addresses all four gaps.

Table 1. Comparison of Agricultural Decision Support Platforms

Platform	Climate	Crop	Price	Free	Target
Cropin	Partial	Yes	No	No	Enterprise
AgriSuite	No	Yes	Partial	No	Commercial
IFFCO Kisan	Yes	Partial	No	Yes	Farmers
Govt. Portals	Yes	No	No	Yes	General
HelloFarmer	Yes	Yes	Yes	Yes	Farmers

All three core capabilities integrated, free, and targeting non-expert farmers.

Methodology

HelloFarmer is organized around three tightly integrated prediction modules: a climate prediction engine, a crop insight engine, and a crop price forecasting module. Each module follows a structured pipeline: data ingestion → feature engineering → model training and validation → real-time inference → user-facing output. The modules share a common data infrastructure built on pandas, NumPy, and PostgreSQL, ensuring consistent preprocessing and reproducibility across all prediction tasks.

Climate Prediction Module

The climate prediction module is responsible for generating 7-day ahead forecasts of daily maximum and minimum temperature, expected rainfall, and relative humidity for the user's registered district. Accurate short-horizon forecasts in these four variables are sufficient to drive downstream crop and price recommendations.

- ***Data Sources and Ingestion:*** Historical daily weather records spanning 20 years (2004–2024) are sourced from IMD station data and supplemented by the OpenWeatherMap historical API for stations with gaps. Records are aligned by station-date key and stored in a time-series table in PostgreSQL. On application startup, the system fetches the last 7 days of observations and appends them to the historical series to ensure the inference window is current.

- ***Feature Engineering:*** Raw time-series features are transformed into a supervised learning format using a sliding window of 30 lag days for each of the four meteorological variables (120 lag features total). Additionally, calendar features are engineered: day-of-year sine/cosine encoding to capture annual seasonality, a binary monsoon flag (June–September), weekly rolling means for temperature and rainfall, and district-level elevation as a static covariate. This yields an input feature vector of 142 dimensions per prediction step.

- ***LSTM Architecture:*** A two-layer stacked LSTM network is trained per district for each target variable. The first LSTM layer has 128 units with return sequences enabled; the second has 64 units. A dropout rate of 0.2 is applied after each LSTM layer to mitigate overfitting. A final Dense layer outputs a single value (point forecast). Models are trained using the Adam optimizer with a learning rate of 0.001, mean squared error loss, and early stopping on a 20% validation split with patience of 10 epochs.

- ***Ensemble Fallback:*** For districts with fewer than 5 years of historical data, an XGBoost Regressor with 200 estimators, max depth 6, and subsample 0.8 serves as a fallback model. This ensemble approach ensures geographic coverage even in data-sparse rural areas. A meta-learner selects between LSTM and XGBoost based on leave-one-year-out cross-validation RMSE, automatically deploying the better-performing model per district.

- ***Suitability Score:*** A composite Agricultural Weather Suitability Score (AWSS) is computed daily as a weighted combination of normalized temperature deviation from crop-optimal range (weight 0.35), rainfall adequacy relative to historical median (0.35), and humidity comfort index (0.30). The score ranges from 0 (highly adverse) to 100 (ideal), and is displayed on the user dashboard as a color-coded daily card.

Crop Insight Engine

The crop insight engine synthesizes outputs from the climate prediction module with user-provided soil parameters to generate ranked crop recommendations and season-specific growing advisories. The engine covers 48 crops commonly cultivated in Maharashtra spanning Kharif, Rabi, and Zaid seasons.

- ***Soil Parameter Input:*** Users input soil NPK levels (kg/ha), pH, organic carbon percentage, and available water capacity through an interactive form. Alternatively, users may upload a soil test PDF from a government-certified laboratory, from which the system extracts values using a regex-based parser. Input values are validated against agronomically plausible ranges and flagged for correction if out of bounds.

- ***Multi-Feature Scoring Algorithm:*** Each candidate crop c is scored using a weighted Euclidean distance metric: Score(c) = 1 / (1 + Σ w_i × |x_i − μ_{c,i}| / σ_{c,i}), where x_i are the user's observed feature values, μ_{c,i} and σ_{c,i} are the crop's optimal mean and tolerance range for feature i, and w_i are weights derived from agronomic literature (soil pH: 0.25; rainfall: 0.25; temperature: 0.20; NPK composite: 0.20; humidity: 0.10). Climate forecast values for the upcoming fortnight are incorporated as dynamic inputs, making recommendations sensitive to near-future conditions rather than only current state.

- ***Seasonality and Sowing Window Filter:*** Each crop entry in the knowledge base includes a sowing calendar with district-specific start and end dates for each season. The engine applies a temporal filter that down-weights crops whose optimal sowing window is more than 3 weeks away and eliminates crops whose window has closed. This prevents the system from recommending climatically suitable crops that are agronomically impractical to sow at the current date.

- ***Pest and Disease Risk Engine:*** An overlay risk model evaluates temperature-humidity combinations against known pathogen and pest activation thresholds for each recommended crop. For example, late blight risk in potatoes is flagged when 5-day average temperature is 10–20°C and humidity exceeds 90%. Risk flags are sourced from the ICRISAT pest management database and displayed as warning banners on the crop recommendation card.

- ***Advisory Content Layer:*** Beyond recommendation, the engine generates growing stage advisories: irrigation scheduling based on evapotranspiration estimates (FAO-56 Penman-Monteith method), fertilizer application timing derived from crop growth-stage models, and harvesting window predictions based on growing degree days (GDD) accumulated from the sowing date. Advisories are presented in plain language with no technical jargon, reviewed for clarity by agricultural officers in our partner extension network.

Crop Price Prediction Module

The price prediction module forecasts 7-day and 14-day ahead modal prices for selected commodities at the user's nearest mandi. The module is critical for helping farmers decide whether to sell immediately, store for later, or transport produce to a more favorable market.

- ***Data Pipeline:*** Historical mandi price data (daily minimum, maximum, and modal prices, arrival quantities, and market location) are sourced from the Agmarknet API and the Maharashtra State Agricultural Marketing Board (MSAMB) open data portal. Data covers 47 commodities across 112 mandis in Maharashtra from 2015 to 2024, totaling approximately 18 million price records. Records are cleaned for duplicates, imputed for sparse markets using spatial interpolation from neighboring mandis, and stored in a partitioned PostgreSQL table indexed by commodity-market-date.

- ***Feature Engineering:*** For each commodity-market pair, the following features are constructed: lag-1, lag-7, lag-14, and lag-30 modal prices; 7-day and 30-day rolling mean and standard deviation of prices; arrival quantity lags and rolling mean (supply proxy); rainfall accumulation in the catchment region (demand-supply interaction); a Fourier-decomposed seasonal component (annual and semi-annual harmonics); and a binary Diwali/Holi/harvest festival calendar flag capturing demand spikes.

- ***XGBoost Model:*** An XGBoost Regressor with 500 estimators, max depth 5, learning rate 0.05, subsample 0.8, and colsample_bytree 0.8 is trained per commodity-market pair. Hyperparameters are tuned using 5-fold time-series cross-validation (walk-forward). The model is retrained weekly as new price data arrives.

- ***Prophet Seasonal Model:*** For commodities with strong annual price cycles (wheat, soybean, sugarcane), a Facebook Prophet model is trained in parallel. Prophet decomposes the price series into trend, yearly seasonality, weekly seasonality, and holiday components. Its output is blended with XGBoost at a 40:60 weight ratio (Prophet:XGBoost) for 14-day forecasts, where seasonal signal becomes more important than lag-price information.

- ***Confidence Intervals and Market Comparison:*** Prediction intervals (80% and 95%) are computed using quantile regression forests trained on the same feature sets. The UI displays a shaded band around the point forecast, communicating uncertainty to the farmer. A Market Comparison View queries forecasts for all mandis within 100 km of the user's location and ranks them by predicted 7-day modal price, enabling informed transport decisions.

System Architecture

HelloFarmer is designed as a modular, scalable web application following a three-tier architecture: a Vue.js 3 Single Page Application (SPA) frontend, a FastAPI asynchronous REST backend, and a PostgreSQL relational database. The ML inference layer is decoupled from the web server using a background task queue, ensuring that long-running model inference does not block API responses. Figure 1 illustrates the high-level architecture.

Frontend Layer

The frontend is built with Vue.js 3 using the Composition API, providing a reactive, component-based user interface. Routing is handled by Vue Router, and state management by Pinia. Chart.js is integrated for all data visualizations including climate forecast line charts, crop suitability radar charts, price trend time-series with confidence bands, and market comparison horizontal bar charts. The frontend communicates with the backend exclusively via RESTful HTTP for CRUD operations and WebSockets for real-time training progress and live weather refresh events.

The interface is fully responsive, designed with a mobile-first approach using CSS Grid and Flexbox. This is essential because a significant proportion of rural farmers access the internet via smartphones. The UI has been localized with i18n support for Marathi and Hindi in addition to English, with a language toggle accessible on every page.

Backend and API Layer

The backend is implemented using FastAPI (Python 3.11), chosen for its native async/await support, automatic OpenAPI documentation generation, and Pydantic-based request validation. All ML inference is executed asynchronously using Python's asyncio and background tasks, preventing the event loop from blocking during model inference. The API exposes the following core endpoint groups: /auth for user registration and JWT-based authentication; /weather for climate forecasts and historical data; /crop for recommendation queries and advisory retrieval; /price for commodity price forecasts and market comparison; and /data for dataset upload and soil parameter management.

A Redis-based caching layer stores pre-computed weather and price forecasts with a TTL of 6 hours, reducing redundant model inference for the same district or commodity-market pair across multiple user requests. Background workers implemented with APScheduler execute model retraining, data ingestion from external APIs, and database vacuuming on configurable schedules.

Database Layer

PostgreSQL (version 15) serves as the primary data store, chosen for its robust support of time-series data through BRIN indexes, JSON column types for flexible metadata storage, and row-level security for user data isolation. The schema comprises six core tables: users (profile and authentication), districts (geographic metadata and station mappings), weather_records (time-series meteorological observations), crop_knowledge (crop parameter vectors and sowing calendars), mandi_prices (historical commodity price series), and user_sessions (interaction history and saved queries). Database migrations are managed using Alembic.

Machine Learning Infrastructure

Trained model artifacts are serialized using joblib and stored in a versioned model registry on the local filesystem, with model metadata (training date, cross-validation scores, feature schema) recorded in a dedicated models table in PostgreSQL. The model loading mechanism uses a lazy-loading singleton pattern, ensuring each model is loaded into memory at most once per worker process. Model versioning enables A/B testing of retrained models against deployed baselines before promotion to production.

Data Flow Summary

A typical user interaction follows a structured workflow. First, the user authenticates into the system and selects their district. The frontend then sends a request for climate forecasting through the /weather/forecast/{district_id} endpoint. The backend checks the Redis cache for previously generated forecasts; if no cached result is available, asynchronous inference is triggered using the district's deployed LSTM model. The generated forecast is then returned to the frontend and stored in the cache for future requests.

Next, the user submits soil parameters to request crop recommendations. The backend combines the climate forecast data with the provided soil inputs, executes the multi-feature scoring engine, and returns a ranked list of suitable crops. When the user selects a commodity, the price forecasting module is activated, and the forecast is either retrieved from the cache or computed dynamically using the hybrid XGBoost–Prophet model. Finally, all user interaction events are recorded in the user_sessions database table to support future personalization and system optimization.

Testing and Validation

HelloFarmer was subjected to a comprehensive testing regime covering unit testing of individual model components, integration testing of inter-module data flows, model validation against held-out datasets, and user interface evaluation with real agricultural stakeholders.

Unit Testing of Prediction Modules

Each prediction module was unit-tested in isolation using pytest. The climate prediction module was tested against synthetic weather series with known statistical properties to verify that LSTM training converges, that lag feature construction is correct, and that the AWSS computation produces values in the expected [0, 100] range. The crop scoring engine was validated using a manually curated set of soil-climate profiles with known agronomically correct crop rankings, verifying that the distance metric produces consistent orderings. The price prediction module was tested with artificially injected price anomalies to confirm that Winsorization preprocessing correctly caps outliers without distorting the price distribution.

The semantic validation pipeline was tested against 50 soil parameter inputs spanning out-of-range pH values, negative NPK entries, and missing fields, confirming that the validation layer correctly flags all invalid inputs and returns informative error messages without crashing the API endpoint.

Model Validation: Climate Prediction

Walk-forward cross-validation was performed on the Maharashtra climate dataset by training on years 2004–2021 and evaluating on 2022 and 2023 separately. The LSTM model achieved a 7-day temperature forecast MAE of 1.8°C and RMSE of 2.4°C across all tested districts. Rainfall forecasting is inherently more uncertain; the model achieved MAE of 7.4 mm and RMSE of 11.2 mm for daily rainfall, with skill score (SS vs. climatological baseline) of 0.43. Humidity forecasts achieved MAE of 5.2% relative humidity. For districts where

XGBoost was deployed as the fallback, temperature MAE was 2.3°C, confirming the LSTM's superiority for data-rich districts.

Model Validation: Crop Recommendation

The crop recommendation engine was evaluated on a held-out test set of 200 farmer survey records from Maharashtra, each containing soil test results, sowing date, and the crop actually cultivated. The engine's top-1 recommendation matched the farmer's actual crop in 71.5% of cases, and top-3 accuracy was 89.2%.

Mismatches were most common for crops grown due to non-agronomic reasons (contract farming, government procurement guarantees), which are outside the engine's current scope. Precision and recall across all 48 crops were computed, with macro-averaged F1-score of 0.78.

Model Validation: Price Forecasting

Price models were validated using walk-forward backtesting on the 2023 calendar year, treating all prior data as training. For tomatoes—among the most volatile crops—the XGBoost model achieved $R^2 = 0.86$ and RMSE of ₹42/quintal on the 7-day forecast horizon. Wheat, with its more stable seasonal pattern, achieved $R^2= 0.91$ and RMSE of ₹28/quintal. The Prophet+XGBoost blend outperformed standalone XGBoost on the 14-day horizon for all crops tested, with an average RMSE reduction of 8.3%. Coverage probability of the 80% prediction interval was 78.4% across all commodity-market pairs, close to the nominal value.

Integration and Performance Testing

Integration tests were written using pytest-httpx to simulate full API call chains from weather forecast requests through crop recommendations to price queries, verifying correct data propagation across module boundaries. Load testing using Locust with 100 concurrent simulated users showed median API response times of 340ms for cached responses and 1.8 seconds for cache-miss inference calls, both within acceptable bounds for web application usability. Database query performance was validated using EXPLAIN ANALYZE on all critical queries, with index coverage confirmed for all filtering and join patterns.

User Interface Evaluation

A usability study was conducted with 30 participants: 15 active farmers from Nashik and Pune districts, and 15 agricultural science students from GSMCOE. Participants were given five standard tasks: (1) view the 7-day climate forecast for their district; (2) input soil parameters and view crop recommendations; (3) read the growing advisory for a recommended crop; (4) check the 14-day price forecast for tomatoes at the nearest mandi; and (5) compare prices across three markets. Task completion rate was 93.3% across all tasks and participants. Mean time on task ranged from 42 seconds (weather forecast) to 3.1 minutes (full soil input and crop recommendation). System Usability Scale (SUS) score averaged 81.4, classified as "Excellent". Qualitative feedback highlighted the weather suitability score card and the market comparison chart as the most practically useful features.

Results and Evaluation

Predictive Model Performance Summary

Table 2 consolidates performance metrics across all three prediction modules evaluated on Maharashtra regional datasets. All metrics are computed on held-out test data using walk-forward or stratified cross-validation as appropriate for each task type.

Table 2. Predictive Model Performance on Held-Out Test Data

Task	Algorithm	Dataset	Metric	Score
Temp. Forecast	LSTM (2-layer)	IMD 2022–23	MAE	1.8°C
Rainfall Forecast	LSTM (2-layer)	IMD 2022–23	RMSE	11.2 mm
Crop Rec. Top-1	Wt. KNN Scoring	Farmer Survey	Accuracy	71.5%
Crop Rec. Top-3	Wt. KNN Scoring	Farmer Survey	Accuracy	89.2%
Price (Tomato, 7d)	XGBoost	Agmarknet 2023	R^2 / RMSE	0.86 / ₹42
Price (Wheat, 7d)	Prophet+XGB	Agmarknet 2023	R^2 / RMSE	0.91 / ₹28
Price (Onion, 7d)	XGBoost	Agmarknet 2023	R^2 / RMSE	0.83 / ₹56

IMD = India Meteorological Department; Agmarknet = National Agricultural Market data portal.

Efficiency and Time-Saving Analysis

A structured observational study compared the time required to complete the full agricultural decision workflow using HelloFarmer against the traditional multi-source approach (consulting IMD website, calling extension officers, visiting mandis, checking newspaper price listings). Ten participants performed both approaches with a two-week washout period between sessions to minimize learning effects. Table III summarizes the results.

Table 3. Workflow Time Comparison: Traditional vs. HelloFarmer

Task	**Algorithm**	**Dataset**	**Metric**	**Score**
Temp. Forecast	LSTM (2-layer)	IMD 2022–23	MAE	1.8°C
Rainfall Forecast	LSTM (2-layer)	IMD 2022–23	RMSE	11.2 mm
Crop Rec. Top-1	Wt. KNN Scoring	Farmer Survey	Accuracy	71.5%
Crop Rec. Top-3	Wt. KNN Scoring	Farmer Survey	Accuracy	89.2%
Price (Tomato, 7d)	XGBoost	Agmarknet 2023	R^2 / RMSE	0.86 / ₹42
Price (Wheat, 7d)	Prophet+XGB	Agmarknet 2023	R^2 / RMSE	0.91 / ₹28
Price (Onion, 7d)	XGBoost	Agmarknet 2023	R^2 / RMSE	0.83 / ₹56

Times are medians across 10 participants. Traditional method includes travel and phone call wait times.

User Satisfaction and Usability

Post-study survey results from the 30-participant usability study showed that 86.7% of participants rated HelloFarmer as "Easy" or "Very Easy" to use without prior training. The weather suitability score was rated the most useful feature by 60% of farmer participants, while the market comparison chart was rated most useful by 73% of student participants. Open-ended feedback identified desired future features: offline mode, voice input in Marathi, and integration with government crop insurance scheme eligibility checks. No critical usability failures were observed; all 30 participants successfully completed at least 4 of the 5 assigned tasks.

Model Export and Reproducibility

All trained models are serialized using joblib with a versioned naming convention (e.g., lstm_nashik_temp_v3_20240901.pkl) and stored in the model registry. Each model artifact includes a companion metadata JSON file recording the training date, training data date range, cross-validation scores, feature schema, and preprocessing pipeline parameters. Validation tests confirmed that deserialized models produce bit-identical predictions to those generated at training time, satisfying reproducibility requirements. The full training pipeline is scripted and parameterized, enabling complete model regeneration from raw data in under 4 hours on a single GPU-equipped server.

Research Gaps and Future Scope

Current Limitations

HelloFarmer in its current form has several acknowledged limitations. First, the platform supports only supervised, structured data inputs; unstructured sources such as satellite imagery, drone photography, or IoT sensor streams are not yet ingested. This limits the granularity of crop health monitoring to what can be inferred from weather and user-reported soil data. Second, the price prediction module's accuracy degrades for highly perishable, volatile crops (e.g., green chillies, coriander) where 7-day price swings of 200–400% are not uncommon and historical patterns provide limited predictive power. Third, the crop recommendation engine does not currently account for water availability constraints—a critical factor in rain-shadow districts such as Marathwada. Fourth, the platform does not yet support multi-user farm management or cooperative-level aggregation, limiting its utility for Farmer Producer Organizations (FPOs).

Identified Research Gaps

A broader survey of the agricultural AI literature identifies several underexplored research directions that are directly relevant to platforms like HelloFarmer. Causal inference in agricultural price modeling remains nascent: most existing approaches are correlational, conflating supply shocks with demand signals and failing to disentangle the causal drivers of price volatility. Methods from the econometrics literature (instrumental variables, structural VAR) have rarely been applied to mandi price data at scale.

Federated learning for privacy-preserving model improvement represents an important open challenge. Farmers are often reluctant to share soil data or yield records due to concerns about data misuse by agribusiness intermediaries. Federated training frameworks that allow model updates without centralizing raw data could unlock vastly larger training datasets while preserving farmer privacy. This is an area where agricultural AI platforms lag significantly behind medical AI, where federated learning is already in production use.

Multi-modal foundation models for agriculture, combining satellite imagery, weather time series, and soil spectral data, are an emerging research frontier. While early prototypes exist in academic settings, their integration into accessible, production web platforms for smallholder contexts remains unexplored.

Future Enhancement Roadmap

Based on identified gaps and user feedback, a concrete roadmap of future enhancements has been prioritized. In the near term (6–12 months): (i) integration of the Sentinel-2 NDVI product via the Copernicus Open Access Hub API for satellite-derived crop stress monitoring; (ii) development of an offline-first Progressive Web App (PWA) with service workers and IndexedDB caching for use in low-connectivity areas; (iii) addition of a Marathi voice interface using the Whisper ASR model for hands-free operation during fieldwork.

In the medium term (12–24 months): (iv) expansion of the price prediction module to incorporate SEBI commodity futures data and export demand indicators from APEDA; (v) implementation of federated learning for the crop recommendation engine, allowing model improvement from opt-in farmer data without centralization; (vi) integration with PM-KISAN beneficiary data and Pradhan Mantri Fasal Bima Yojana (PMFBY) crop insurance APIs to provide scheme eligibility and claim guidance directly within the platform.

In the long term (24+ months): (vii) development of a multi-tenant cooperative management module supporting Farmer Producer Organizations with shared land parcel maps, bulk input procurement optimization, and collective selling coordination; (viii) extension of the climate prediction module to seasonal outlooks (3–6 month) using statistical downscaling of coupled climate model (CMIP6) outputs, enabling farmers to plan crop rotations ahead of the upcoming growing season.

Conclusion

This paper presented HelloFarmer, a comprehensive web-based agricultural decision support platform that integrates climate prediction, crop insight generation, and commodity price forecasting into a unified, accessible interface designed for non-expert farmers and agricultural stakeholders. The platform was motivated by the persistent information asymmetry between what modern agricultural data science can provide and what smallholder farmers can actually access in practice.

Three tightly integrated ML modules were designed, implemented, and validated: an LSTM-based climate forecasting engine achieving temperature MAE of 1.8°C; a multi-feature crop scoring engine achieving top-3 recommendation accuracy of 89.2% against ground-truth farmer records; and an XGBoost/Prophet price forecasting system achieving R^2 scores of 0.83–0.91 across three major Maharashtra commodities. The platform's Vue.js frontend, FastAPI backend, and PostgreSQL database provide a scalable, responsive architecture that maintained sub-2-second response times under realistic concurrent load.

User evaluation with 30 farmers and agricultural students confirmed an average System Usability Scale score of 81.4 ("Excellent") and a task completion rate of 93.3% without prior training. Time-savings analysis demonstrated a 96% reduction in total workflow time compared to traditional multi-source information gathering. These results validate both the technical efficacy and practical accessibility of the HelloFarmer platform.

The broader significance of this work lies in demonstrating that the integration gap in agricultural AI—where powerful models exist in silos, inaccessible to the farmers who need them most—can be closed through thoughtful systems engineering and user-centered design. HelloFarmer is released as an open-access platform, and the authors invite collaboration from agricultural research institutions, government bodies, and the farming community to extend its capabilities toward the enhancement roadmap outlined in Section VII.

Acknowledgment

The authors express sincere gratitude to Assistant Professor Vatsala Priya and staff of the Department of Artificial Intelligence and Data Science, Genba Sopanrao Moze College of Engineering, Pune, for providing the computational infrastructure and academic environment that made this research possible. We thank our project supervisor for sustained guidance throughout the design and development phases. Appreciation is extended to the open-source communities behind Vue.js, FastAPI, scikit-learn, XGBoost, and Facebook Prophet, whose frameworks form the foundation of this platform. We are particularly grateful to the 15 farmers from Nashik and Pune districts and the 15 students from GSMCOE who participated in our usability evaluation, and to the agricultural extension officers of the Maharashtra State Agriculture Department who reviewed the advisory content for technical accuracy. Finally, we acknowledge the India Meteorological Department and the Maharashtra State Agricultural Marketing Board for making meteorological and mandi price data publicly accessible.

References

1. M. S. Agarwal, P. Singh, and R. Tiwari, "Crop recommendation system using machine learning: A survey," Int. J. Comput. Appl., vol. 182, no. 15, pp. 1–8, 2019.
2. S. Agrawal and N. Agrawal, "Survey on anomaly detection using data mining techniques," Procedia Comput. Sci., vol. 60, pp. 708–713, 2015.
3. P. Sharma, R. Kumar, and A. Singh, "Agricultural commodity price prediction using machine learning approaches," Int. J. Comput. Appl., vol. 180, no. 12, pp. 15–22, 2018.
4. India Meteorological Department, "Gridded Rainfall and Temperature Data Products," IMD, New Delhi, 2024. [Online]. Available: https://imdpune.gov.in
5. OpenWeatherMap, "Weather API Documentation," 2024. [Online]. Available: https://openweathermap.org/api
6. Ministry of Agriculture and Farmers Welfare, "Agricultural Statistics at a Glance 2023," Govt. of India, 2023.
7. Z. Zakaria, M. N. Sulaiman, and R. Mustapha, "Machine learning framework for precision agriculture advisory systems," J. Agric. Technol., vol. 45, no. 3, pp. 231–248, 2023.
8. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Disc. Data Mining, pp. 785–794, 2016.
9. S. J. Taylor and B. Letham, "Forecasting at scale," Am. Stat., vol. 72, no. 1, pp. 37–45, 2018.
10. R. Elshawi, M. Maher, and S. Sakr, "Automated machine learning: State-of-the-art and open challenges," ACM Comput. Surv., vol. 54, no. 2, pp. 1–37, 2021.
11. F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.
12. S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.
13. Vue.js Core Team, "Vue.js 3 — The Progressive JavaScript Framework," 2024. [Online]. Available: https://vuejs.org
14. S. Ramirez, "FastAPI: Modern, fast web framework for building APIs with Python," 2024. [Online]. Available: https://fastapi.tiangolo.com
15. A. K. Nair, S. Prasad, and V. Menon, "Deep learning approaches for crop yield prediction: A review," Comput. Electron. Agric., vol. 193, p. 106701, 2022.
16. J. Brooke, "SUS: A quick and dirty usability scale," in Usability Evaluation in Industry, P. W. Jordan et al., Eds. London: Taylor & Francis, 1996, pp. 189–194.