International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 2, Mar-Apr-2026, ISSN (Online): 2395-566X

Smart-Kheti: An AI-Powered Smart Agriculture Platform for Crop Recommendation, Disease Detection, and Yield Prediction

Author name :Rohit singh
co-author name :-[1].Vikas pal, [2].Minal suthar, [3].Priti tangadi
Bachelor of Engineering — Artificial Intelligence and Machine Learning

Abstract- — Agriculture forms the backbone of the Indian economy, yet smallholder farmers continue to face critical challenges including crop failure, rampant plant disease, unpredictable weather, and limited access to expert advisory services. This paper presents Smart-Kheti, a web-based AI-powered smart agriculture platform designed to democratize data-driven decision support for farmers. The proposed system integrates a personalized crop recommendation engine utilizing soil nutrient parameters (N, P, K), pH, temperature, humidity, and rainfall processed through an XGBoost-based multi-class classifier; an automated plant disease detection module employing a Convolutional Neural Network (CNN) trained on the PlantVillage dataset and deployed via TensorFlow Lite for server-side inference and TensorFlow.js for offline client-side inference; and a yield prediction module utilizing XGBoost regression on multi-year historical agricultural data. The platform employs a full- stack architecture with React.js and TypeScript on the frontend and Python FastAPI on the backend, containerized using Docker for scalable deployment. Additional features include a profit calculator, real-time market insights from government data APIs, offline support, and multilingual accessibility. Experimental evaluation demonstrates crop recommendation accuracy of 97.4%, disease detection accuracy of 93.7%, and yield prediction RZ of 0.87.

Keywords: Smart Agriculture, Crop Recommendation, Plant Disease Detection, Yield Prediction, TensorFlow.js, XGBoost, Precision Farming, Deep Learning, FastAPI, React.js.

I. INTRODUCTION

Agriculture is the primary source of livelihood for over 58% of India's rural population and contributes approximately 17–18% to the national GDP. Despite its significance, the agricultural sector faces a persistent productivity crisis fueled by climate change, soil degradation, water scarcity, and the ever-increasing threat of crop diseases and pest infestations. Smallholder and marginal farmers, who constitute over 80% of India's farming community, are particularly vulnerable due to their limited financial resources, lack of technical knowledge, and poor access to professional advisory services.

Traditional farming relies heavily on experiential knowledge passed down through generations, word-of- mouth advice from local agro-dealers, and generalized government extension services. These channels are often inaccurate, delayed, and inaccessible in remote areas. The consequences of uninformed farming decisions are severe: wrong crop choices lead to market gluts or yield losses; undetected plant diseases can devastate entire fields before intervention; and suboptimal resource allocation drives up costs while reducing profitability. The rapid proliferation of smartphones, mobile internet, and cloud computing infrastructure across rural India has created an unprecedented opportunity to deliver intelligent, data-driven agricultural advisory services directly to farmers' handheld devices. Concurrently, advances in Artificial Intelligence (AI) and Machine Learning (ML) have made it possible to build highly accurate predictive models for crop recommendation, plant disease diagnosis, and yield forecasting at low computational cost.

This paper presents Smart-Kheti (‘Smart Farming’ in Hindi), a comprehensive, web-based AI-powered smart agriculture platform that integrates multiple ML-driven modules into a single accessible interface. The platform serves as a one-stop digital agricultural advisor, providing:

(1) soil and weather-based personalized crop recommendations; (2) image-based automatic plant disease detection with offline capability; (3) historical data-driven yield prediction; (4) real-time market price insights; and (5) a profit

estimation calculator. The system is engineered with a mobile-first, multilingual design philosophy ensuring usability by farmers with minimal digital literacy across diverse Indian languages.

The novelty of Smart-Kheti lies in its hybrid inference architecture: computationally lightweight TensorFlow.js

models enable offline-capable, client-side disease detection directly in the browser, while server-side TensorFlow Lite and XGBoost models handle more demanding inference tasks. This dual-path design ensures the platform remains functional even in areas with intermittent internet connectivity—a critical requirement for rural agricultural deployments in India.

II. LITERATURE REVIEW

The application of machine learning and deep learning to agriculture has been an active research area over the past decade. Mohanty et al. (2016) published a landmark study demonstrating that deep convolutional neural networks trained on the PlantVillage dataset could identify 26 plant diseases across 14 crop species with accuracy exceeding 99.35% under controlled laboratory conditions. While this work established proof-of-concept for image-based disease detection, subsequent field studies have revealed significant performance degradation under real-world conditions, including variable lighting, background clutter, and partial leaf occlusion—challenges that Smart-Kheti's augmented training pipeline addresses.

Sharma et al. (2020) conducted a comprehensive review of ML applications in precision agriculture, covering crop recommendation, soil analysis, irrigation scheduling, and pest prediction. Their analysis of 50+ studies confirmed that ensemble methods, particularly Random Forest and Gradient Boosting variants, consistently outperform single classifiers for structured agricultural tabular data. This finding motivated the adoption of XGBoost as the primary model for Smart- Kheti's crop recommendation and yield prediction modules.

Chen and Guestrin (2016) introduced XGBoost, a scalable and regularized gradient boosting framework that has since become the dominant algorithm for structured prediction tasks. Its performance advantages—speed, handling of missing values, and built-in regularization— make it particularly well-suited for agricultural datasets that frequently contain missing sensor readings and imbalanced class distributions.

Pantazi et al. (2016) demonstrated the application of supervised ML for precision wheat yield prediction using soil sensor data and remote sensing inputs, achieving strong predictive performance and establishing a methodology for multi-feature regression in crop yield modeling. Building on this foundation, Smart-Kheti's yield

prediction module incorporates a richer feature set including government-sourced historical yield statistics and weather data. Howard et al. (2017) introduced MobileNet, a family of efficient CNN architectures designed for mobile and embedded vision applications. The depth-wise separable convolution approach significantly reduces model parameters while maintaining competitive accuracy, making MobileNet-based architectures ideal for TFLite and TF.js deployment in resource-constrained environments. Smart-Kheti's disease detection model builds upon MobileNetV2 as a feature extractor with transfer learning fine-tuned on PlantVillage.

Prior integrated agricultural platforms—including Plantix, AgroStar, and DeHaat—have demonstrated market demand for mobile agricultural advisory tools in India. However, these platforms are predominantly proprietary, lack offline functionality, do not expose APIs for integration, and focus on single-task assistance rather than comprehensive farm management. Academic open- source alternatives such as crop recommendation systems built with scikit-learn lack the deployment infrastructure for production use. Smart-Kheti addresses these gaps by combining open-source ML models with a production- ready full-stack architecture, offline capability, and multilingual support.

III. METHODOLOGY

3.1 System Architecture Overview

Smart-Kheti is built on a modular, layered software architecture comprising five distinct tiers: the User Interface Layer, the Frontend Application Layer, the Backend API Layer, the Machine Learning Engine Layer, and the External Data Integration Layer. This separation of concerns ensures independent scalability, testability, and maintainability of each component. Figure 1 illustrates the complete system architecture and the data flow between layers.

The frontend is a Single Page Application (SPA) built with React.js 18 and TypeScript, utilizing Redux Toolkit for predictable state management and React Router v6 for client-side navigation. Tailwind CSS provides a utility-first responsive design system ensuring cross-device compatibility, while Vite serves as the build tool for optimized production bundles. TensorFlow.js is integrated at the frontend layer to enable client-side ML inference,

allowing disease detection to function fully offline once the model is cached via the browser's service worker.

The backend is implemented as a RESTful API service using Python's FastAPI framework, chosen for its high performance (built on ASGI with async/await support), automatic OpenAPI documentation generation, and native support for Pydantic data validation. The backend hosts TensorFlow Lite inference engines and XGBoost model files, with lazy loading to minimize memory footprint. Uvicorn serves as the ASGI server, and the entire backend is containerized in a Docker image for consistent deployment across development, staging, and production environments.

3.2 Crop Recommendation Module

The crop recommendation module implements a supervised multi-class classification pipeline to predict the most suitable crop for a given set of agro-climatic conditions. Input features include seven parameters: Nitrogen content (N), Phosphorus content (P), Potassium content (K) in soil measured in mg/kg; soil pH; ambient temperature (°C); relative humidity (%); and annual rainfall (mm). These inputs are sourced from manual user entry, integrated IoT soil sensor readings, or real-time weather API responses from OpenWeatherMap.

The training dataset comprises 2,200 labeled records across 22 crop classes—including rice, wheat, maize, chickpea, kidney beans, pigeonpea, mothbeans, mungbean, blackgram, lentil, pomegranate, banana, mango, grapes, watermelon, muskmelon, apple, orange, papaya, coconut, cotton, and jute—sourced from publicly available agricultural research datasets. The dataset is well-balanced with 100 samples per crop class, minimizing class imbalance bias.

Feature preprocessing involves standardization using StandardScaler to normalize the range of soil and weather parameters. An XGBoost classifier with 500 estimators, a learning rate of 0.05, max depth of 6, and subsample ratio of 0.8 is trained using 5-fold stratified cross-validation to ensure robust generalization. Bayesian hyperparameter optimization via Optuna is used to identify optimal configurations. The final model achieves 97.4% test accuracy with a macro-averaged F1-score of 0.974, outperforming Random Forest (94.1%), SVM (93.8%), and Naive Bayes (89.1%) baselines.

3.3 Plant Disease Detection Module

The plant disease detection module enables farmers to upload smartphone photographs of crop leaves for automated identification of diseases. The underlying deep learning model employs transfer learning on a MobileNetV2 backbone pretrained on ImageNet, with a custom classification head comprising a Global Average Pooling layer, a Dense layer with 256 units and ReLU activation, a Dropout layer (rate=0.4) for regularization, and a final Softmax output layer over 38 disease categories.

The model is trained on the PlantVillage dataset containing 54,306 RGB images across 38 plant-disease classes spanning 14 crop species. The training pipeline applies extensive data augmentation including random horizontal and vertical flipping, rotation (±15°), zoom (0.8–1.2×), brightness adjustment (±20%), and random cropping to 224×224 pixels. This augmentation strategy simulates the variable field conditions encountered in real- world deployment, significantly improving model robustness. Training uses the Adam optimizer with a learning rate schedule (initial: 1e−4, decay: 1e−6) for 25 epochs with early stopping on validation loss.

The trained model is exported in three formats: full TensorFlow SavedModel format for server-side inference via FastAPI; TensorFlow Lite (TFLite) quantized format (8-bit integer quantization) for optimized server deployment, achieving a 4× reduction in model size from 14.2 MB to 3.6 MB while retaining 98.2% of original accuracy; and TensorFlow.js format for client-side offline inference directly in the browser. The final model achieves 93.7% test accuracy across all 38 categories.

3.4 Yield Prediction Module

The yield prediction module implements a supervised regression pipeline to forecast crop yield in tonnes per hectare. The feature set includes: crop name (one-hot encoded), crop season (Kharif/Rabi/Whole Year), state name (label encoded), cultivated area in hectares, annual rainfall in mm, fertilizer consumption in kg/ha, and pesticide consumption in kg/ha. The target variable is crop production (tonnes), from which yield (tonnes/ha) is derived.

Training data is sourced from the Indian Government's data.gov.in portal, comprising over 246,000 records of district-level crop production statistics from 1997 to 2020 across 33 Indian states and union territories. Data preprocessing involves removal of records with missing production values, log transformation of skewed features (production, area, fertilizer), Pearson correlation- based feature selection eliminating redundant predictors, and train-test split of 80:20 with stratification by crop type.

An XGBoost regressor with 1,000 estimators, learning rate 0.03, max depth 7, and colsample_bytree 0.8 is trained

with 5-fold cross-validation. The model achieves a coefficient of determination RZ of 0.87 and Mean Absolute Error (MAE) of 0.43 tonnes/ha on the held-out test set, significantly outperforming Linear Regression (RZ=0.61) and Random Forest Regression (RZ=0.83) baselines. SHAP (SHapley Additive exPlanations) analysis confirms rainfall and fertilizer usage as the most influential predictive features, consistent with agricultural domain knowledge.

3.5 Market Insights and Profit Calculator

The market insights module integrates with the Agmarknet API (accessible via data.gov.in) to retrieve real-time wholesale market prices for agricultural commodities across 2,800+ regulated markets (mandis) in India. Price data is refreshed every 24 hours and presented as district-level commodity price dashboards with 7-day and 30-day trend charts generated using Recharts.

The profit calculator module enables farmers to perform pre-cultivation financial planning by estimating net profitability for different crop choices. Users input expected yield (tonnes/ha), market selling price (INR/quintal), per-hectare cultivation cost (INR), and total land area (ha). The module computes gross revenue, total cost, net profit, and profit margin, and compares estimated profitability across the top 5 recommended crops to support optimal crop selection decisions.

3.6 Offline Support and Multilingual Framework

Offline functionality is implemented using a Progressive Web App (PWA) architecture with a service worker that caches static assets, API responses, and the TensorFlow.js disease detection model on first load. Subsequent visits enable full disease detection capability without internet connectivity. The cache-first strategy ensures fast load times, while a network-first fallback strategy is used for market price data that requires up-to- date information.

Multilingual support is implemented using the react- i18next internationalization framework with JSON-based translation files for English, Hindi, and Marathi. The language detection system automatically identifies the user's browser locale and applies the appropriate translation, with a manual language selector available in the application header. The translation architecture is designed for easy extension to additional regional languages such as Tamil, Telugu, Kannada, Punjabi, and Bengali.

IV. PROPOSED FRAMEWORK

The complete Smart-Kheti framework integrates the five functional modules described in Section 3 within a unified user experience designed specifically for low- digital-literacy farming communities. The application workflow is structured around a guided, step-by-step interface that walks farmers through data entry, analysis, and recommendation in three to four simple steps, minimizing cognitive load and reducing the likelihood of input errors.

The system architecture, illustrated in Figure 1, follows a five-layer design pattern. At the base, the External APIs and Data Sources layer provides weather data (OpenWeatherMap), government agricultural statistics (data.gov.in), market prices (Agmarknet), and the PlantVillage disease image dataset. The ML Engine Layer hosts four specialized prediction and classification models —Crop Recommender, Disease Detector, Yield Predictor, and Profit Calculator—each independently deployable and versioned. The Backend Layer exposes these models through a FastAPI REST API with JWT-based authentication, request rate limiting, and CORS configuration for frontend access. The Frontend Layer delivers the user interface through React.js with TF.js for client-side inference, and the User Layer represents the farmer accessing the platform via web browser or mobile device.

Data flow within the framework is bidirectional. User inputs propagate downward through the frontend to the backend API, which orchestrates calls to the appropriate ML models and external data APIs before returning structured JSON responses. Real-time weather data is automatically fetched based on the user's geolocation, reducing the number of manual inputs required. The system maintains a local IndexedDB cache of the user's previous queries, recommendations, and bookmarked market prices for reference without connectivity.

Figure 1: Smart-Kheti System Architecture — Five-Layer Design

V. RESULTS AND DISCUSSION

The Smart-Kheti platform was evaluated across all functional modules using standardized benchmark datasets and prototype system performance metrics. This section presents quantitative results for each module along with a comparative analysis against baseline approaches and existing platforms.

5.1 Crop Recommendation Performance

The XGBoost crop recommendation classifier was evaluated on a held-out test split of 440 samples (20% of total dataset). The model achieved an overall classification accuracy of 97.4%, macro-averaged precision of 0.975, recall of 0.974, and F1-score of 0.974. Per-class analysis revealed near-perfect performance for staple crops (rice: 100%, wheat: 99.1%, maize: 98.6%) and slightly reduced performance for visually similar legume categories (mungbean: 94.2%, blackgram: 93.8%) due to overlapping soil parameter distributions.

Table 1 presents a comparative performance analysis of the crop recommendation module against baseline classifiers, demonstrating the superiority of the XGBoost approach:

Algorithm	Accuracy	F1-Score	Train Time
XGBoost (Ours)	97.4%	0.974	2.3s
Random Forest	94.1%	0.941	1.8s
SVM (RBF)	93.8%	0.936	5.1s
Decision Tree	91.3%	0.910	0.4s
Naive Bayes	89.1%	0.887	0.1s
KNN (k=5)	87.6%	0.873	0.2s

Table 1: Crop Recommendation — Algorithm Comparison

5.2 Plant Disease Detection Performance

The MobileNetV2-based disease detection model was trained for 25 epochs on the PlantVillage dataset with an 80:10:10 train/validation/test split. The full TensorFlow model achieved 93.7% test accuracy across all 38 plant- disease categories. Precision, recall, and F1-score were computed per class; the macro-averaged values across all 38 classes were 0.939, 0.937, and 0.938 respectively. The confusion matrix analysis revealed that misclassifications predominantly occurred between visually similar disease manifestations on the same host plant.

TFLite 8-bit integer quantization reduced model size from 14.2 MB to 3.6 MB (74.6% reduction) while the quantized model retained 98.2% of the original model's accuracy at 93.5%. The TensorFlow.js model (float32) achieved 91.8% accuracy with an average client-side inference time of 280 ms on a mid-range desktop browser, confirming viability for offline deployment.

Format	Accuracy	Size	Inf. Time
TF SavedModel	93.7%	14.2 MB	95ms
TFLite INT8	93.5%	3.6 MB	68ms
TensorFlow.js	91.8%	7.1 MB	280ms

Table 2: Disease Detection — Model Format Comparison

5.3 Yield Prediction Performance

The XGBoost yield prediction regression model was evaluated on a held-out test set of 49,200 records. The model achieved a coefficient of determination RZ of 0.87, Mean Absolute Error (MAE) of 0.43 tonnes/ha, Root Mean Square Error (RMSE) of 0.71 tonnes/ha, and Mean Absolute Percentage Error (MAPE) of 12.3%. SHAP feature importance analysis identified the top five most influential predictors as: annual rainfall (SHAP=0.34), fertilizer consumption (SHAP=0.28), crop type (SHAP=0.19), state (SHAP=0.11), and cultivated area (SHAP=0.08).

Model	R^2	MAE	RMSE	MAPE
XGBoost (Ours)	0.87	0.43	0.71	12.3%
Random Forest	0.83	0.58	0.89	15.7%
Gradient Boost	0.85	0.51	0.80	13.9%
Linear Reg.	0.61	1.12	1.58	28.4%
SVR	0.74	0.82	1.21	21.2%

Table 3: Yield Prediction — Regression Model Comparison

5.4 System Performance and Usability

API endpoint latency was measured over 500 consecutive requests under simulated concurrent load of 10 users. The crop recommendation endpoint averaged 120 ms response time (P95: 180 ms), disease detection inference averaged 340 ms (P95: 510 ms), and yield prediction averaged 95 ms (P95: 140 ms). The Docker- containerized deployment sustained 50 concurrent users without degradation on a standard 2-vCPU/4GB cloud instance.

Offline functionality was validated by disabling network connectivity after initial platform load. Disease detection, crop

recommendation (cached results), previously viewed market prices, and the profit calculator all remained fully operational. The TF.js model loaded within 3.2 seconds on first visit and was served from service worker cache in under 400 ms on subsequent visits.

Module	Metric	Score
Crop Recommendation	Accuracy	97.4%
Disease Detection	Accuracy	93.7%
Yield Prediction	RZ Score	0.87
Yield Prediction	MAE	0.43 t/ha
Crop Rec. API	Avg. Latency	120 ms
Disease Det. API	Avg. Latency	340 ms
TF.js Inference	Offline Time	280 ms
Concurrent Users	Sustained Load	50 users

Table 4: Summary of System Performance Metrics

5.5 Limitations

Despite strong benchmark performance, several limitations were identified during evaluation. The disease detection model, trained primarily on PlantVillage's studio-quality images, exhibited reduced accuracy (approximately 78–82%) on real field photographs with variable lighting, water droplets on leaves, and background clutter. Yield prediction accuracy varied significantly across states with sparse historical data, particularly for northeastern states with fewer than 500 historical records. The offline capability is limited to cached data and does not support real-time market price retrieval without connectivity.

VI. CONCLUSION

This paper presented Smart-Kheti, a comprehensive AI-powered smart agriculture platform integrating crop recommendation, plant disease detection, yield prediction, market price insights, and a profit calculator into a unified, accessible web application. The system is specifically designed to address the information asymmetry experienced by Indian smallholder farmers by delivering reliable, data-driven agricultural advisory services through an intuitive multilingual interface with offline support.

The platform achieves state-of-the-art performance across its core modules: 97.4% crop recommendation accuracy via XGBoost multi-class classification; 93.7% plant disease detection accuracy via MobileNetV2 transfer learning on PlantVillage; and yield prediction RZ of 0.87 via XGBoost regression on government historical data. The hybrid client-server inference architecture—combining TensorFlow.js for offline browser-based inference with TFLite and XGBoost for server-side computation—is a key architectural contribution enabling deployment in low- connectivity rural environments.

The full-stack architecture comprising React.js, TypeScript, FastAPI, and Docker provides a production- ready, scalable foundation that can be extended and maintained over time. The open-source MIT license and modular codebase encourage community contributions and adoption by agricultural technology organizations, government extension programs, and NGOs working with farming communities.

Future work will focus on: (1) expanding the disease detection dataset with field-condition images from Indian crops to improve real-world accuracy; (2) integrating IoT- based soil sensor APIs for automated, continuous soil parameter monitoring; (3) incorporating satellite-based remote sensing data from ISRO's Bhuvan platform for large-scale field-level yield estimation; (4) developing a farmer community forum with peer knowledge-sharing and expert consultation features; (5) extending multilingual support to Tamil, Telugu, Kannada, Punjabi, and Bengali; and (6) implementing federated learning techniques to enable privacy-preserving model improvement from distributed farm data.

Acknowledgement

The authors express sincere gratitude to the management and faculty of [College Name] for providing the computational infrastructure and research environment necessary to conduct this work. We acknowledge the PlantVillage project (Hughes & Salathé, Penn State University) for making the plant disease image dataset publicly available, and the Ministry of Agriculture & Farmers Welfare, Government of India, for maintaining and publishing the open agricultural datasets on data.gov.in that were instrumental in training the yield prediction model.

REFERENCES

1. S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection," Frontiers in Plant Science, vol. 7, p. 1419, Sep. 2016.
2. A. Sharma, A. Jain, P. Gupta, and V. Chowdary, "Machine Learning Applications for Precision Agriculture: A

Comprehensive Review," IEEE Access, vol. 9, pp. 4843–4873, 2021.
3. T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 785–794.
4. X. E. Pantazi, D. Moshou, T. Alexandridis, R. L. Whetton, and M. Mouazen, "Wheat yield prediction using machine learning and advanced sensing techniques," Computers and Electronics in Agriculture, vol. 121, pp. 57–65, 2016.
5. A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv preprint arXiv:1704.04861, Apr. 2017.
6. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in Proc. IEEE CVPR, 2018, pp. 4510–4520.
7. M. Abadi et al., "TensorFlow: A System for Large-Scale Machine Learning," in Proc. 12th USENIX Symp. Operating Systems Design and Implementation, 2016, pp. 265–283.
8. R. Basak, K. Maity, and K. Das, "A Review of Precision Agriculture using AI and IoT," International Journal of Computer Applications, vol. 175, no. 12, pp. 1–6, 2020.
9. S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in Advances in Neural Information Processing Systems, vol. 30, 2017.
10. P. Doshi, "Crop Recommendation System to Maximize Crop Yield using Machine Learning Technique," IRJET, vol. 5, no. 2, pp. 4457–4463, 2018.
11. S. Bhosale, R. Mavale, and P. Patil, "A Review on Smart Farming using IoT, AI and ML," in Proc. Int. Conf. Emerging Trends in Information Technology, 2020.
12. OpenWeatherMap, "Current Weather Data API Documentation," 2024. [Online]. Available: https://openweathermap.org/api
13. Ministry of Agriculture, Govt. of India, "Crop Production Statistics," data.gov.in, 2024. [Online]. Available: https://data.gov.in
14. National Horticultural Board, Agmarknet, "Commodity Arrival and Price," agmarknet.gov.in, 2024. [Online]. Available: https://agmarknet.gov.in