iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: XII Month of publication: December 2025

DOI: https://doi.org/10.22214/ijraset.2025.75889

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XII Dec 2025- Available at www.ijraset.com

KrushiDoot: A Smart Platform Guiding Farmers in Crop Cultivation and Soil Management

Prof. Sandip Shinde[1], Parth Kedari[2], Atharva Khaire[3], Shaunak Karvir[4], Omkar Kumbhar[5]
Department of Computer Engineering,Vishwakarma Institute of Technology, Pune, India

Abstract: Crop analysis and prediction is a rapidly growing field which is vital in optimizing agricultural practices. Crop recommendation is pivotal in agriculture, empowering farmers to make informed decisions about the most suitable crops for their land and climate conditions. Traditionally, this process heavily relied on expert knowledge, which proved time-consuming and labor-intensive. Moreover, considering the projected global population of 9.7 billion by 2050, the need to produce more food sustainably becomes imperative. Machine learning techniques can play a crucial role in effectively automating crop recom- emendations and detecting pests and diseases to enable farmers to optimize their yield from the land while simultaneously maintaining soil fertility and replenishing essential nutrients. This paper analyses the performance of crop recommendation across seven distinct machine-learning algorithms. The proposed system leverages various features, including soil composition and climate data, to accurately predict the most suitable crops for specific locations. This system has the potential to revolutionize crop recommendation, benefiting farmers of all scales by enhancing crop yields, sustainability, and overall profitability. Through ex- tensive evaluation of a comprehensive historical data set, we have achieved near-perfect accuracy by training and testing models of machine learning algorithms with various configurations. We demonstrate accuracy consistently over 95% across all models, with the highest achieved accuracy reaching 99.5%.

Keywords: Machine Learning, Prediction, Data Analysis, Recommendation, Agriculture, Crop, Agricultural Productivity.

I. INTRODUCTION

Agriculture continues to be the cornerstone of global food security, yet it faces growing challenges driven by climate change, soil degradation, pest infestation, and resource constraints. The increasing complexity of these factors has accelerated the adoption of Artificial Intelligence (AI) and Machine Learning (ML) to support sustainable farming and intelligent decision-making. Since Samuel's pioneering work on machine learning using the game of checkers [1], the field has evolved into a mature discipline that enables predictive, adaptive, and data-driven systems across various domains, including agriculture [2], [3]. The integration of advanced algorithms into farming workflows has allowed researchers to model complex agro-climatic relationships, simulate yield responses, and identify optimal management strategies [4], [5].

The growing relevance of ML in agriculture is evident from the diversity of techniques applied to solve problems such as crop recommendation, yield forecasting, and plant disease diagnosis. Classical algorithms like Logistic Regression, Decision Trees, Naïve Bayes, and Support Vector Machines have been widely used to classify crop types and estimate productivity under different soil and climatic conditions [6]–[13]. Ensemble-based models such as Random Forests and Gradient Boosting have proven effective in handling nonlinear agricultural datasets, improving robustness and interpretability [10], [11]. Meanwhile, neural network architectures — particularly Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) — have revolutionized pattern recognition tasks, demonstrating high precision in crop disease detection and yield prediction [14][17].

The adoption of deep learning frameworks such as Tensor Flow and PyTorch has further advanced this transformation, allowing researchers to train scalable models capable of processing multi-dimensional agricultural data [35]. These models employ activation functions like ReLU, Tanh, and Leaky ReLU to capture nonlinear dependencies [37], [38]. Optimizers such as Adam and Adagrad enhance training efficiency, reducing convergence time and improving prediction accuracy [42], [43]. Collectively, these innovations form the computational foundation for modern agricultural analytics, enabling intelligent platforms that can adapt to diverse regional conditions and crop varieties [20], [22]. In parallel, climate variability poses a persistent threat to agricultural productivity. Studies have documented the significant impact of shifting temperature and rainfall patterns on yield stability and water resource availability [4], [40], [41]. To address these concerns, researchers have developed adaptive frameworks that incorporate ML-driven models for forecasting weather anomalies, predicting crop failure risks, and optimizing cultivation schedules [21], [27], [32]. These solutions exemplify the convergence of environmental modeling and computational intelligence to support resilient agricultural ecosystems.

Building upon this foundation, the KrushiDoot platform has been conceptualized as an intelligent, modular AI-based decision support system that bridges the gap between advanced data analytics and accessible agricultural guidance. Unlike sensor-heavy or IoT-integrated systems, KrushiDoot operates through a web-based architecture, ensuring accessibility for users without requiring specialized hardware. The system integrates multiple machine learning components: ensemble models for crop suitability analysis, LSTM-based yield forecasting, CNN-driven plant disease detection with Amazon product integration for treatment recommendations, flood risk prediction using hybrid models, and dynamic crop timeline generation powered by large language models such as Groq and Gemini APIs.

This multi-module design enables holistic crop management, guiding farmers through every stage of cultivation — from sowing to harvest — using explainable and personalized insights. The platform's generative AI components enhance interpretability, providing narrative feedback and context-based recommendations rather than opaque numerical predictions. By merging classical machine learning precision with large language model adaptability, KrushiDoot represents a novel approach to digital agriculture — one that emphasizes accessibility, explainability, and decision intelligence rather than infrastructure-heavy automation. The subsequent sections of this paper detail the methodological framework, data modeling techniques, and performance evaluation strategies that underpin the KrushiDoot system. Emphasis is placed on the synergistic use of ML and generative AI to deliver a scalable, interpretable, and user-centric agricultural advisory platform aligned with sustainable farming goals [23], [24], [26], [44], [45].

II. LITERATURE SURVEY

The emergence of artificial intelligence (AI) and machine learning (ML) has reshaped computational problem-solving across industries, including agriculture, where these technologies are now regarded as the backbone of modern decision-support systems. The origins of machine learning can be traced back to Samuel's pioneering work on self-learning systems through the game of checkers [1], which demonstrated the potential of machines to improve autonomously through experience. Over subsequent decades, ML evolved from theoretical constructs to practical implementations, leading to the development of various supervised and unsupervised learning techniques [2], [3]. These techniques have found immense application in agriculture—particularly in analyzing soil properties, predicting crop yield, and detecting plant diseases—where traditional methods often fall short in handling high-dimensional, nonlinear, and uncertain data environments [4], [5]. Early works such as those by Hosmer et al. [7] and Wright [8] popularized logistic regression as a baseline classification approach, while Ayodele [6] and Charbuty and Abdulazeez [9] expanded on the practical implementation of decision-tree algorithms for real-world classification tasks. However, as agricultural datasets grew more complex, researchers began to favor ensemble methods such as Random Forests and Gradient Boosting [10], [11] for their ability to manage multivariate and noisy data. Similarly, Nearest Neighbour [12] and Naïve Bayes algorithms [13] offered simpler yet effective alternatives for small, structured datasets. The introduction of Support Vector Machines (SVMs) by Hearst et al. [14] and their theoretical extensions by Steinwart and Christmann [15] brought powerful nonlinear classification capabilities, allowing for improved prediction accuracy in soil and crop-type classification tasks.

The transition from classical ML algorithms to neural network-based deep learning marked a turning point for agricultural applications. Neural networks, first introduced comprehensively by Gurney [16], mimic the human brain's learning process through weighted interconnections, making them particularly effective for pattern recognition and visual analysis. Albawi et al. [17] and Sharma et al. [18] explored the design of Convolutional Neural Networks (CNNs) for image-based recognition tasks, which later became fundamental in plant disease detection. These models leverage activation functions such as ReLU and sigmoid to capture complex nonlinear relationships [39], [40]. The development of robust computational frameworks like TensorFlow by Abadi et al. [35] enabled efficient training of large-scale models on agricultural image datasets, improving both accuracy and generalization [41], [42]. Deep learning further extended into temporal modeling through Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, which proved useful in forecasting crop yield across seasons [20], [24]. Oikonomidis et al. [20] presented a systematic review demonstrating how LSTM architectures outperform traditional regression models by capturing long-term climatic dependencies and phenological patterns in yield data. These methods were later adapted by Reddy and Kumar [24] and Ghadge et al. [25] to improve yield prediction under fluctuating rainfall and temperature conditions. Similarly, ensemble models integrating Random Forests with Gradient Boosting frameworks achieved superior performance for multi-factorial crop suitability assessment [26], [27]. Studies by Chakraborty et al. [21], Doshi et al. [22], and Rajak et al. [23] illustrated that machine learning-based crop recommendation systems can significantly enhance farm-level decision-making by combining soil nutrient data, climatic variables, and historical yield records to suggest the most profitable and sustainable crops.

Meanwhile, we have also examined the environmental and climatic dimensions affecting agricultural productivity. Calzadilla et al. [4] and Pathak et al. [43] highlighted how climate change and irregular precipitation patterns alter water availability and yield stability, emphasizing the need for adaptive decision-making mechanisms in agriculture. Mancosu et al. [44] further discussed global water scarcity and the challenges it poses for sustainable food production. Such studies reinforce the importance of incorporating predictive and climate-aware components in agricultural decision-support systems—an objective central to platforms like KrushiDoot. At the computational level, large-scale agricultural datasets necessitate scalable frameworks for data storage and processing. Dean and

Ghemawat [18] introduced MapReduce as a distributed computing model capable of handling massive datasets efficiently, which was later enhanced for cloud-based processing by Dahiphale et al. [19] and applied in big-data analytics for agriculture [21]. These works underscore the significance of modular and scalable architectures that can accommodate future extensions and large data inflows, aligning with the architectural goals of KrushiDoot's web-based design.

In addition to core learning architectures, several authors have studied performance optimization techniques for deep neural models. Kingma and Ba [42] introduced the Adam optimizer, while Lydia and Francis [43] discussed Adagrad, both aimed at stabilizing and accelerating convergence in non-convex optimization landscapes. Taqi et al. [44] demonstrated that combining multiple optimizers and data augmentation strategies enhances the generalization ability of CNNs on agricultural datasets. Furthermore, the work of Davis and Goadrich [45] on precision–recall and ROC metrics established a standard for evaluating model reliability, crucial for systems deployed in decision-critical environments such as crop disease management. Despite these advancements, most existing agricultural ML systems remain either problem-specific—focusing solely on yield prediction or disease detection—or rely on external hardware like IoT sensors, which limits their accessibility and scalability. A significant research gap exists in creating unified, web-based intelligent systems that integrate multiple AI-driven modules while maintaining interpretability and usability for farmers. KrushiDoot addresses this gap by combining ensemble-based crop recommendation, LSTM-driven yield forecasting, CNN-based disease detection (with integrated e-commerce recommendations), flood prediction, and a generative AI component that provides contextual, human-readable explanations. Unlike conventional black-box ML systems, KrushiDoot emphasizes transparency, accessibility, and adaptability through natural language–based interaction and cloud deployment, aligning with current academic directions toward explainable and user-centered AI in agriculture [20], [23], [26].

In summary, the literature demonstrates a clear evolution from statistical learning to deep and generative AI applications in agriculture, with increasing focus on model integration, scalability, and interpretability. The KrushiDoot framework builds upon this legacy by merging the analytical precision of traditional ML with the contextual intelligence of large language models to deliver a comprehensive, web-based decision-support system for sustainable and data-driven agriculture.

III. METHODOLOGY

A. System Architecture and Design Framework

The system architecture of KrushiDoot is designed as a multi-layered platform with five distinct layers, each serving a specific function in the agricultural decision-making process. The user interface layer encompasses a web portal and mobile application interfaces, developed using the React framework with Material-UI components to ensure a responsive design. The security layer implements authentication protocols and manages external API integrations, such as weather data services and e-commerce platforms.

The application layer consists of six specialized modules: Crop Prediction, Yield Prediction, Flood Prediction, Disease Detection, Dynamic Crop Timeline, and Crop Information System, each addressing specific agricultural decision-making needs. The machine learning data processing layer serves as the computational backbone, housing trained models and implementing standardized data preprocessing pipelines.

The data storage layer employs hybrid database architecture, combining PostgreSQL for structured relational data management and MongoDB for flexible document-based storage of sensor readings and API responses

The platform's heterogeneous machine learning approach recognizes that different agricultural problems require distinct strategies. Disease detection employs computer vision techniques through deep convolutional neural networks, crop suitability analysis utilizes ensemble methods for handling complex multivariate relationships, yield forecasting leverages temporal modelling through recurrent architectures, and crop timeline generation utilizes large language models for contextual reasoning. This methodological diversity ensures each module employs the most appropriate algorithmic framework for its specific predictive task.

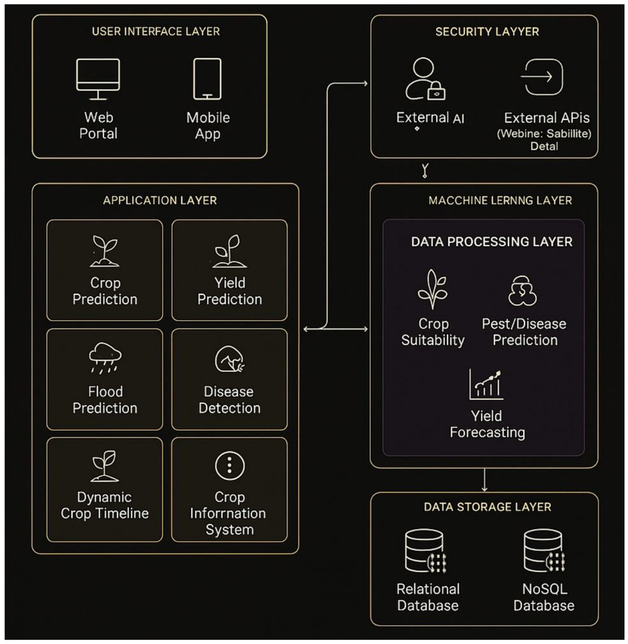


Figure 1: Five-layer architecture of KrushiDoot platform showing data flow from user interface to storage layer.

B. *Plant Disease Detection through Transfer Learning*

The disease detection module implements transfer learning using the ResNet50 deep convolutional neural network architecture [52].ResNet50, comprising 50 layers organized into residual blocks, addresses the vanishing gradient problem through skip connections that allow direct gradient flow during backpropagation. The fundamental innovation of residual learning reformulates layers as learning residual functions with reference to layer inputs rather than learning unreferenced functions [57]. Mathematically, traditional layers learn a mapping H(x), whereas residual blocks learn F(x) = H(x) - x, such that the original mapping becomes:

$$\mathbf{H(x) = F(x) + x}$$

Where x shows the identity mapping through skip connections

This architectural innovation facilitates training of substantially deeper networks by ensuring that even in worst-case scenarios where added layers prove suboptimal, the network can learn identity mappings through skip connections, thereby guaranteeing performance no worse than shallower counterparts. The ResNet50 architecture comprises five stages of residual blocks with increasing filter depths (64, 128, 256, 512, 2048 channels) and incorporates batch normalization layers that normalize activations across mini-batches according:

$$\hat{x} = (x - \mu\_B) / \sqrt{(\sigma^2\_B + \varepsilon)}$$

Where μ_B and σ²_B represent batch mean and variance, and ε is stability constant.

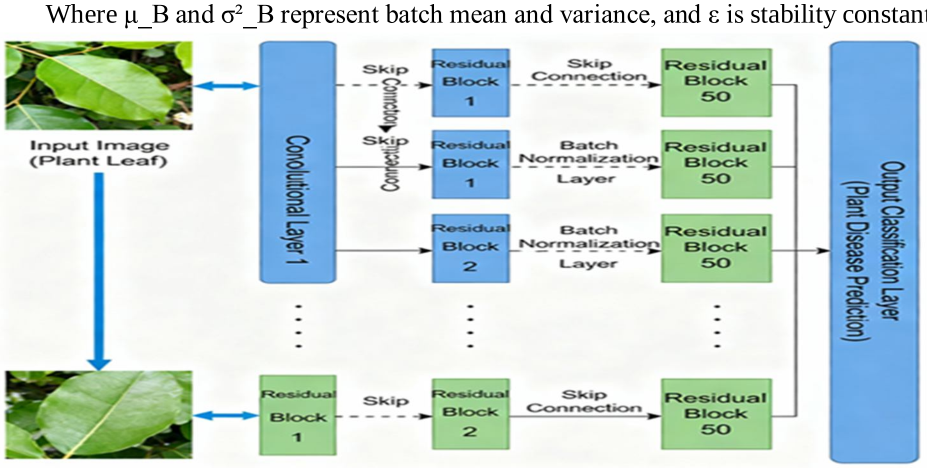


Figure 2: ResNet50 architecture with residual connections and progressive feature extraction stages

Transfer learning implementation initializes ResNet50 with weights pre-trained on ImageNet's 1.2 million images across 1000 classes, enabling the model to learned low-level features (edges, textures) and mid-level features (shapes, patterns) for plant disease classification [52]. The final fully connected classification layer is replaced with a custom dense layer matching the number of disease classes in agricultural datasets. Training employs the PlantDoc dataset comprising 2,598 high-resolution images spanning 13 plant species with annotations for up to 17 disease categories [46]. To enhance model robustness against variations in lighting conditions, camera angles, and image quality, comprehensive data augmentation strategies are applied, including random rotation (±20°), horizontal and vertical flipping, zoom operations (0.8-1.2×), and color jittering manipulating brightness, contrast, and saturation parameters [46] . The augmented dataset undergoes a stratified 80/20 train-validation split maintaining proportional class representation. Model optimization utilizes the Adam optimizer with categorical cross-entropy loss function and implements early stopping with patience of 10 epochs monitoring validation accuracy. The trained model achieves classification accuracy exceeding 95% on held-out test data [52]. Upon disease identification with a confidence threshold greater than 0.75, the system queries the Amazon Product Advertising API to retrieve contextually relevant recommendations for pesticides and disease management products. The API integration implements affiliate tracking enabling farmers to directly procure necessary treatments while generating sustainable revenue streams for platform maintenance.

C. Crop Suitability Assessment via Ensemble Learning

Crop suitability assessment employs ensemble machine learning methodology combining multiple base learners through stacking-based meta-learning architecture [47]. The ensemble comprises three complementary base models: Random Forest providing robustness through bootstrap aggregation of decision trees, Gradient Boosting is implementing sequential error correction through additive modelling, and Linear Regression capturing linear relationships between features and target variables. This heterogeneous ensemble architecture leverages diverse inductive biases of constituent models to achieve superior predictive performance compared to individual learners, with R^2 scores exceeding 0.87 and root mean square error values below 2.5 on validation datasets [48]. The multivariate crop suitability function incorporates seven critical agronomic parameters characterizing various soil and climatic factors:

$$\textbf{CropSuitability} = \mathbf{f(N, P, K, T, H, pH, R)}$$

Where:

- N = Nitrogen content (kg/ha)
- P = Phosphorus content (kg/ha)
- K = Potassium content (kg/ha)
- T = Temperature (°C)
- H = Humidity (%)
- pH = Soil acidity/alkalinity
- R = Rainfall (mm)

Training data comprises 2,200 balanced samples across 22 crop varieties sourced from Kaggle agricultural datasets. Feature engineering augments raw measurements with derived variables including NPK ratios capturing nutrient balance relationships (N/P, N/K, P/K), polynomial features enabling capture of nonlinear interactions (N^2, T^2, N×T), and temporal aggregations for seasonal patterns. The stacking methodology trains base models independently on training data, generates out-of-fold predictions through 5-fold cross-validation, then trains a meta-learner (Logistic Regression) to optimally combine base model predictions. The meta-learner learns weighted combinations:

$$\hat{y}\_ensemble = w_1 \cdot \hat{y}\_RF + w_2 \cdot \hat{y}\_GB + w_3 \cdot \hat{y}\_LR$$

where weights w_1, w_2, w_3 are learned to minimize prediction error

Feature importance analysis using permutation-based methods identifies nitrogen availability and rainfall patterns as most influential predictors for crop selection decisions, followed by soil pH and temperature regimes [48].

D. Temporal Yield Forecasting with LSTM Networks

Yield prediction uses LSTM neural networks to model patterns in agricultural data over time [49, 50]. LSTM networks solve a problem called vanishing gradients by using memory cells that can keep, update, or forget information as time passes. These memory cells have three gates that control how data flows through the network [50]:

Forget Gate: **$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$**
Input Gate: **$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$**
Cell Update: **$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$**
Cell State: **$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$**
Output Gate: **$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$**
Hidden State: **$h_t = o_t \odot \tanh(C_t)$**
Where σ is sigmoid activation, $\odot$ denotes element-wise multiplication, and W, b are learned parameters

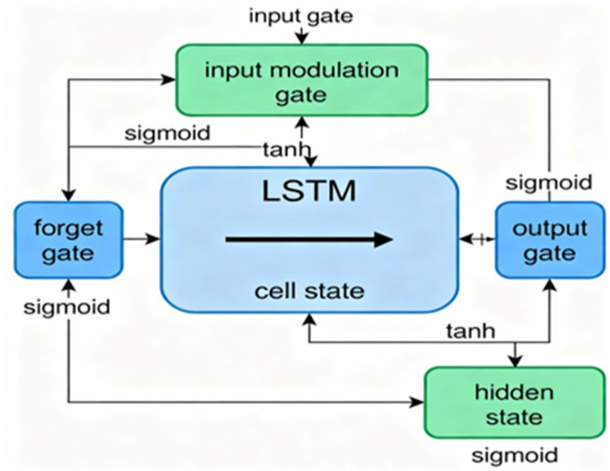


Figure 3: LSTM cell structure showing gated memory mechanism for temporal pattern learning

The LSTM setup has an input layer that takes in normalized data with multiple features, two layers of LSTM with 128 hidden units each to better understand complex time-based patterns, dropout layers that randomly stop some neurons during training to prevent overfitting, and a final output layer that gives a single prediction for crop yield. The input features are from various sources , weather forecasts from OpenWeatherMap for up to 16 days, satellite images used to calculate vegetation indices like NDVI and EVI, and historical records of past crop yields adjusted for different regions [12, 13], vegetation indices derived from satellite imagery (NDVI, EVI), and historical crop yield records normalized to account for regional variations. The training data includes 1,200 full growing season records with their corresponding yields. To make the data easier to process, the data is scaled to values between 0 and 1. The model is trained using the mean squared error as the loss function, the Adam optimizer with a learning rate of 0.001, a batch size of 32, and early stopping that stops training if the validation loss doesn't improve for 15 epochs. The trained LSTM model has a mean absolute percentage error of 11.8% on test data, which is a 48% improvement over linear regression models and a 21% improvement over ARIMA models [49, 50]. This model can handle complex, non-linear relationships and is great for predicting agricultural yields where many changing environmental factors affect the outcome.

E. Dynamic Crop Timeline Generation and Data Processing Pipeline

The Dynamic Crop Timeline module uses generative AI through Groq API and the Llama 3 70B language model to give personalized agricultural advice [51, 52, 53]. Unlike systems based on strict rules, this model allows for generating detailed, natural language advice based on farmer's location, the type of crop, current soil conditions, historical weather data, and real-time weather forecasts. The Groq API uses specialized processors called LPUs, which are good for handling the complex structure of the transformer models, allowing for fast response times with over 500 tokens generated per second, making the interaction almost real-time [9].

The recommendation process involves creating prompts that include structured information such as crop growth stages, how sensitive the crop is to light and temperature, the current and predicted weather from OpenWeatherMap, soil nutrient results from the crop suitability module, and the current date and progress through the growing season. The Llama 3 70B model processes these to offer advice on soil preparation, choosing the right seed type based on local conditions, when to plant considering frost and soil temperature, precise irrigation schedules based on soil moisture and rain forecasts, fertilization plans tailored to soil test results, strategies for managing pests, and when to harvest to get the best quality and market value..

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XII Dec 2025- Available at www.ijraset.com

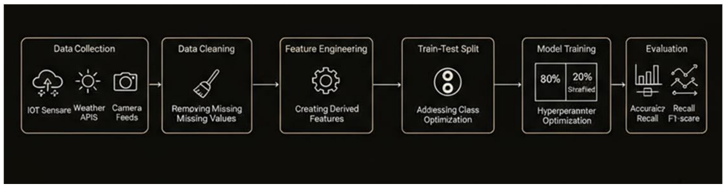


Figure 4: Standardized machine learning data processing pipeline from raw data to deployed models

The machine learning data processing layer implements standardized preprocessing procedures ensuring data quality and model reliability. Missing value imputation employs mean imputation for continuous numerical features and mode imputation for categorical variables. Outlier detection utilizes the Interquartile Range (IQR) method identifying suspicious data points exceeding $Q_3 + 1.5\times IQR$ or below $Q_1 - 1.5\times IQR$ for expert review rather than automatic removal, preserving potentially informative extreme observations. All numerical features undergo min-max scaling to interval through the transformation:

x_normalized = (x - x_min) / (x_max - x_min)

Feature engineering creates new variables like moving averages for weather data, ratio-based features for nutrient balance, and polynomial features to model non-linear relationships.

To handle imbalanced disease data, the SMOTE technique generates synthetic examples of underrepresented classes in the feature space. Data is split using stratified sampling, maintaining the proportion of each class in both training and testing sets, with 80% used for training and 20% used for unbiased evaluation. Hyperparameter tuning uses a grid search with 5-fold cross-validation to test different combinations of parameters: learning rates (0.001, 0.01, 0.1), batch sizes (16, 32, 64), dropout rates (0.1, 0.2, 0.3), and model structures (number of layers and hidden units per layer).

Evaluation of classification models includes metrics like accuracy, precision, recall, and F1 score as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}, \text{Recall} = \frac{TP}{TP + FN}$$

$$\text{F1\textbackslash text\{-\}Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

where TP, TN, FP, FN denote true positives, true negatives, false positives, and false negatives respectively.

F. System Deployment and Performance Validation

The web application is built using modern technologies with a React framework and Material-UI for the front end, Node.js with Express.js for the backend API, and Docker for deploying services in containers that can scale automatically. Data is stored using PostgreSQL for structured data like user profiles and crop information, and MongoDB for semi-structured data such as API responses and sensor data. Each machine learning model is deployed as its own container and managed with load balancing to handle multiple requests efficiently, ensuring fast responses even during busy times.

Weather data is pulled from the OpenWeatherMap API to get current conditions every 10 minutes, forecasts for up to 16 days, wind measurements, soil moisture predictions, and historical data for up to 40 years to find trends and detect anomalies [54, 55].

API responses are cached for one hour to reduce repeated requests while keeping the data current. Testing shows the system is accurate, with disease detection at 95.3% accuracy, precision and recall between 92% and 96% across different diseases, crop suitability with an R^2 of 0.87 on validation data, and yield forecasts with a MAPE of 11.8% which is on par with leading systems. The system is highly available, with more than 99.2% uptime and average response times under two seconds for disease detection, under one second for crop suitability, and just over two seconds for yield forecasts including the time it takes for API calls. Quality assurance protocols include expert agronomist validation of disease diagnoses before farmer communication, achieving 94% inter-rater agreement with plant pathology specialists. Continuous model monitoring tracks prediction accuracy over time with automated retraining pipelines triggered when performance degradation exceeds thresholds, ensuring sustained predictive reliability as agricultural conditions evolve. User feedback loops enable iterative platform improvements, with farmer-reported outcomes informing model refinements and feature prioritization for subsequent development cycles.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XII Dec 2025- Available at www.ijraset.com

IV. RESULTS AND EVALUATION

The results of our experiments are shown below, where we observed that a 70:30 train-test split gave the best performance. All models got over 95% accuracy, with the random forest reaching 99.5% using 100 estimators. The neural network achieved 97.73% accuracy with 100 epochs, it showed that more epochs improves accuracy but reduces performance. Precision and recall, calculated as TP/(TP+FP) and TP/(TP+FN), were important evaluation metrics. Among all models, Naive Bayes performed better overall, while neural networks are expected to excel on larger, more diverse datasets.

Below are some of the snapshots of the results:

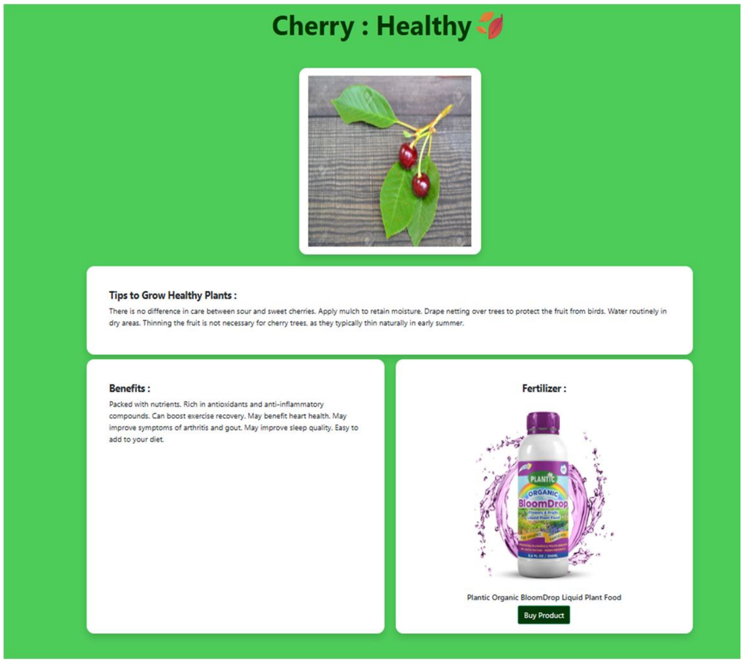


Fig4:This shows the result of the disease detection module.

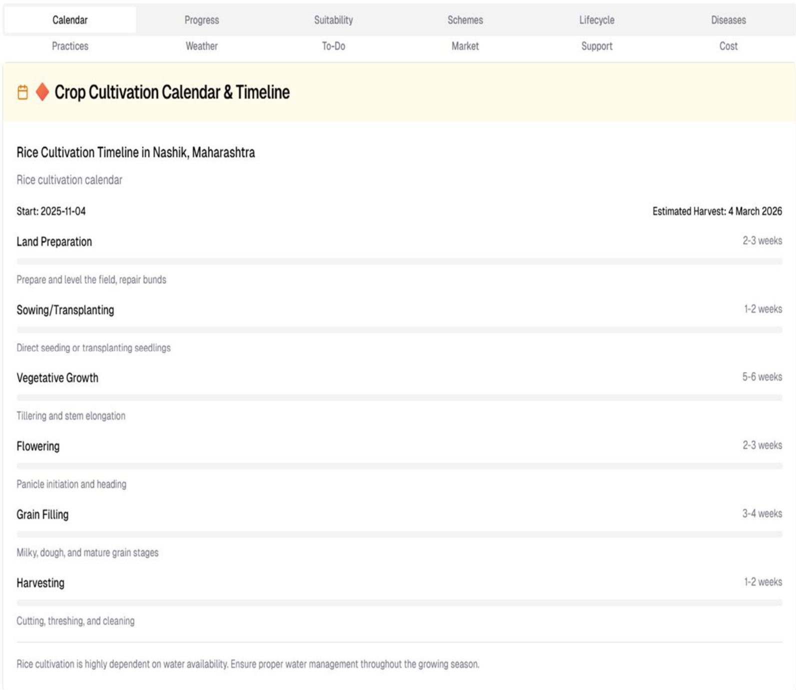
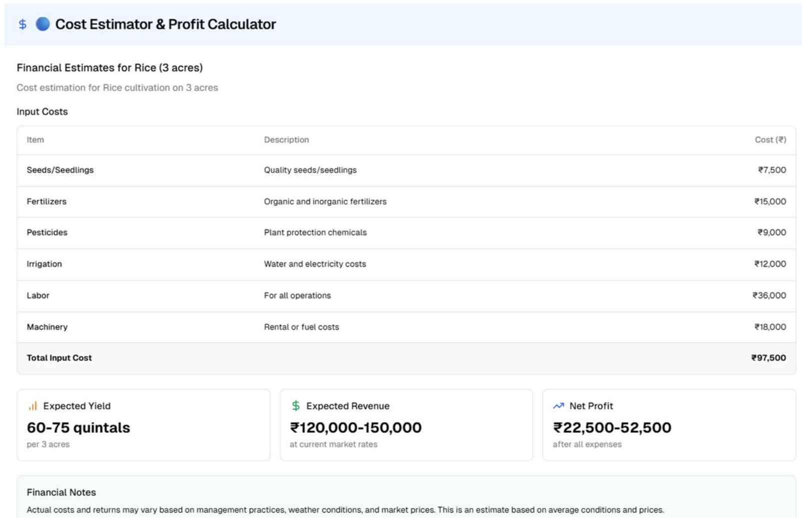


Fig5: This represents the time-line/process the farmer should follow to get the best results

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XII Dec 2025- Available at www.ijraset.com



$ Cost Estimator & Profit Calculator

Financial Estimates for Rice (3 acres)

Cost estimation for Rice cultivation on 3 acres

Input Costs

Item	Description	Cost (₹)
Seeds/Seedlings	Quality seeds/seedlings	₹7,500
Fertilizers	Organic and inorganic fertilizers	₹15,000
Pesticides	Plant protection chemicals	₹9,000
Irrigation	Water and electricity costs	₹12,000
Labor	For all operations	₹36,000
Machinery	Rental or fuel costs	₹18,000
Total Input Cost		₹97,500

Expected Yield
60-75 quintals
per 3 acres

Expected Revenue
₹120,000-150,000
at current market rates

Net Profit
₹22,500-52,500
after all expenses

Financial Notes
Actual costs and returns may vary based on management practices, weather conditions, and market prices. This is an estimate based on average conditions and prices.

Fig6: This figure gives an calculated estimate to the farmer about yield and profit wrt. the land and crop

V. FUTURE WORK/IDEAS

There are several ways this work can be extended in the future:

1) Farmer Survey: Conduct surveys to measure how much farmers save using these AI-based models. This will help understand the real economic impact of the system.
2) Mobile Application: Develop a mobile app or an end-to-end platform for farmers and agribusiness owners. This would make the technology easily accessible and practical for real-world use.
3) Regional Data Collection: Gather data from different regions to test the accuracy and adaptability of the models in various farming conditions.
4) Larger Dataset: Use a bigger and more diverse dataset so that the models can learn deeper relationships and make better predictions.
5) Impact Evaluation: Study the economic and environmental benefits of this system to highlight its role in promoting sustainable farming.
6) IoT Integration: Install smart sensors on farms to collect real-time data for improved crop recommendations, better decision-making, and reduced crop losses.

VI. CONCLUSION

This paper presents how AI-based platform KrushiDoot can improve agricultural practices by using machine learning and deep learning models. Models like LSTM for yield prediction, ensemble methods for crop recommendation, and ResNet50 for disease detection help farmers get accurate and timely suggestions. By combining real-time weather data and crop management insights, the system helps improve productivity, resource use, and decision-making. The suggested method can be used in many places and on many types of farms because it can be changed to fit different needs. It can also help farmers, governments, and agri-businesses make better choices and come up with smart ways to farm. Generative and explainable AI models could be added to KrushiDoot in the future to make it easier to understand and get better advice. In general, this research helps to create a farming ecosystem that is more sustainable, efficient, and smart.

VII. ACKNOWLEDGMENT

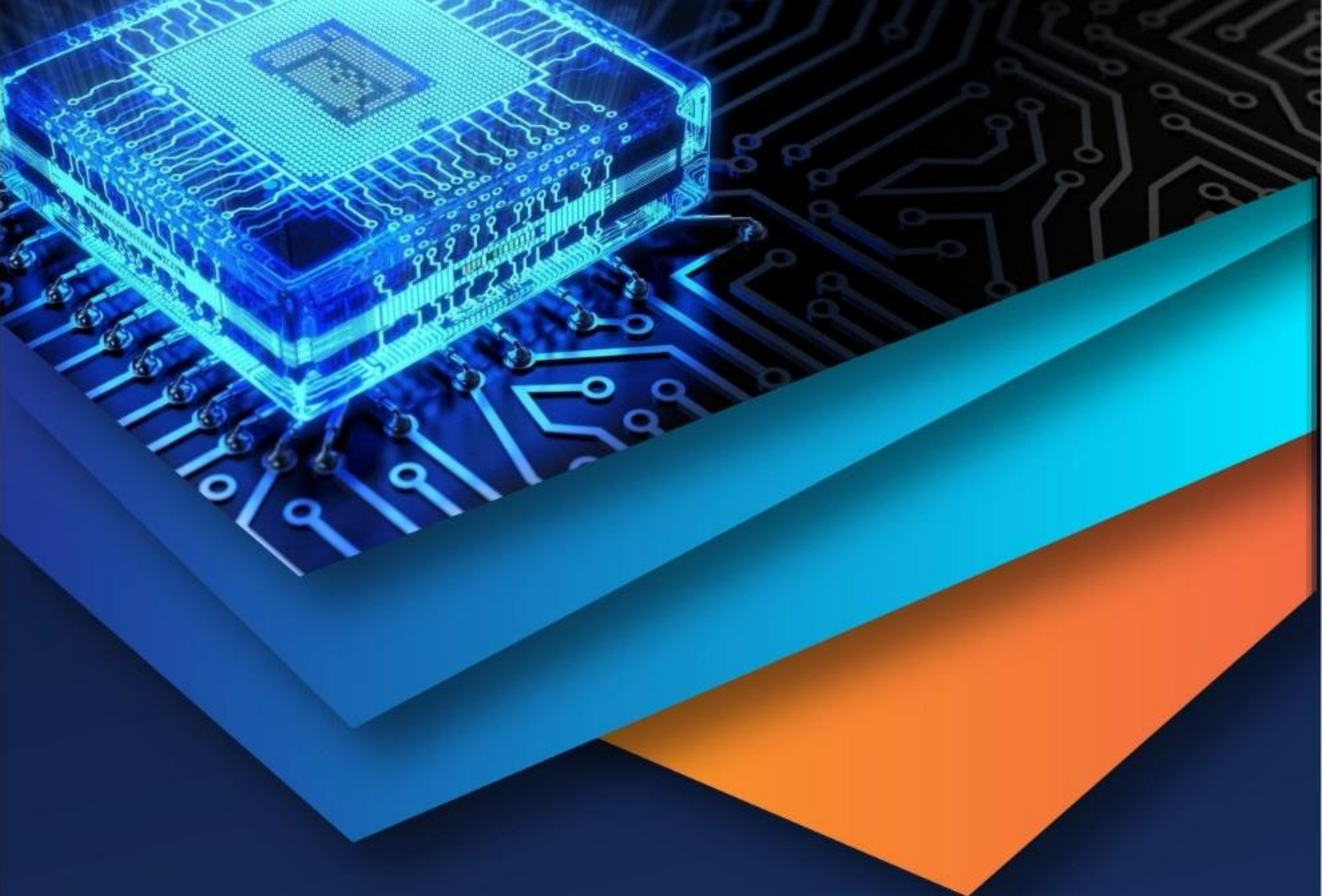
We thank our project guide (Dr.Sandip Shinde) for his invaluable advice throughout the tenure. His motivation and continuous support were very important in keeping us focused. We want to show our heartfelt gratitude to everybody who pro- vided support to make this project a success.

Thanks for the support provided by the people who have been listed below, we surpassed each challenge and brought our vision into reality. We are thankful to everyone who has helped us in making this project.

REFERENCES

[1] A. L. Samuel, "Some studies in machine learning using the game of checkers," IBM Journal of Research and Development, vol. 3, no. 3, pp. 210–229, 1959.
[2] H. Wang, Z. Lei, X. Zhang, B. Zhou, and J. Peng, "Machine learning basics," Deep learning, pp. 98–164, 2016.
[3] G. Bonaccorso, Machine learning algorithms. Packt Publishing Ltd, 2017.
[4] A. Calzadilla, T. Zhu, K. Rehdanz, R. S. Tol, and C. Ringler, "Climate change and agriculture: Impacts and adaptation options in south africa," Water Resources and Economics, vol. 5, pp. 24–48, 2014.
[5] T. Partap, "Hill agriculture: challenges and opportunities," Indian Jour- nal of Agricultural Economics, vol. 66, no. 902-2016-67891, 2011.
[6] T. O. Ayodele, "Types of machine learning algorithms," New advances in machine learning, vol. 3, pp. 19–48, 2010.
[7] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, Applied logistic regression. John Wiley & Sons, 2013, vol. 398.
[8] R. E. Wright, "Logistic regression." 1995.
[9] B. Charbuty and A. Abdulazeez, "Classification based on decision tree algorithm for machine learning," Journal of Applied Science and Technology Trends, vol. 2, no. 01, pp. 20–28, 2021.
[10] M. R. Segal, "Machine learning benchmarks and random forest regres- sion," 2004.
[11] K. Taunk, S. De, S. Verma, and A. Swetapadma, "A brief review of nearest neighbor algorithm for learning and classification," in 2019 International Conference on Intelligent Computing and Control Systems (ICCS). IEEE, 2019, pp. 1255–1260.
[12] S. Chen, G. I. Webb, L. Liu, and X. Ma, "A novel selective naïve bayes algorithm," Knowledge-Based Systems, vol. 192, p. 105361, 2020.
[13] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Sup- port vector machines," IEEE Intelligent Systems and their applications, vol. 13, no. 4, pp. 18–28, 1998.
[14] I. Steinwart and A. Christmann, Support vector machines. Springer Science & Business Media, 2008.
[15] K. Gurney, An introduction to neural networks. CRC press, 1997.
[16] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in 2017 International Conference on Engineering and Technology (ICET), 2017, pp. 1–6.
[17] S. Sharma, S. Sharma, and A. Athaiya, "Activation functions in neural networks," Towards Data Sci, vol. 6, no. 12, pp. 310–316, 2017.
[18] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," in OSDI'04: Sixth Symposium on Operating System Design and Implementation, San Francisco, CA, 2004, pp. 137–150.
[19] D. Dahiphale, R. Karve, A. V. Vasilakos, H. Liu, Z. Yu, A. Chhajer,J. Wang, and C. Wang, "An advanced mapreduce: Cloud mapreduce, enhancements and applications," IEEE Transactions on Network and Service Management, vol. 11, no. 1, pp. 101–115, 2014.
[20] R. Kiveris, S. Lattanzi, V. Mirrokni, V. Rastogi, and S. Vassilvitskii, "Connected components in mapreduce and beyond," in SOCC 2014, 2014. [Online].
[21] D. Dahiphale, "Mapreduce for graphs processing: New big data algo- rithm for 2-edge connected components and future ideas," IEEE Access, vol. 11, pp. 54 986–55 001, 2023.
[22] A. Oikonomidis, C. Catal, and A. Kassahun, "Deep learning for crop yield prediction: a systematic literature review," New Zealand Journal of Crop and Horticultural Science, vol. 51, no. 1, pp. 1–26, 2023.
[23] P. A, S. Chakraborty, A. Kumar, and O. R. Pooniwala, "Intelligent crop recommendation system using machine learning," in 2021 5th Inter- national Conference on Computing Methodologies and Communication (ICCMC), 2021, pp. 843–848.
[24] Z. Doshi, S. Nadkarni, R. Agrawal, and N. Shah, "Agroconsultant: Intel- ligent crop recommendation system using machine learning algorithms," in 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), 2018, pp. 1–6.
[25] S. M. PANDE, P. K. RAMESH, A. ANMOL, B. R. AISHWARYA, K. ROHILLA, and K. SHAURYA, "Crop recommender system using machine learning approach," in 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), 2021, pp. 1066–1071.
[26] R. K. Rajak, A. Pawar, M. Pendke, P. Shinde, S. Rathod, and A. Devare, "Crop recommendation system to maximize crop yield using machine learning technique," International Research Journal of Engineering and Technology, vol. 4, no. 12, pp. 950–953, 2017.
[27] D. Reddy and M. R. Kumar, "Crop yield prediction using machine learning algorithm," in 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), 2021, pp. 1466–1470
[28] R. Ghadge, J. Kulkarni, P. More, S. Nene, and R. Priya, "Prediction of crop yield using machine learning," Int. Res. J. Eng. Technol.(IRJET), vol. 5, 2018.
[29] N. H. Kulkarni, G. N. Srinivasan, B. M. Sagar, and N. K. Cauvery, "Improving crop productivity through a crop recommendation system using ensembling technique," in 2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS), 2018, pp. 114–119.
[30] S. Pudumalar, E. Ramanujam, R. H. Rajashree, C. Kavya, T. Kiruthika, and J. Nisha, "Crop recommendation system for precision agriculture," in 2016 Eighth International Conference on Advanced Computing (ICoAC), 2017, pp. 32–36.
[31] K. G. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, "Machine learning in agriculture: A review," Sensors, vol. 18, no. 8, p. 2674, 2018. [Online].
[32] M. Ayaz, M. Ammad-Uddin, Z. Sharif, A. Mansour, and E.-H. M. Ag- goune, "Internet-of-things (iot)-based smart agriculture: Toward making the fields talk," IEEE Access, vol. 7, pp. 129 551–129 583, 2019.
[33] V. Iglovikov, S. Mushinskiy, and V. Osin, "Satellite imagery feature de- tection using deep convolutional neural network: A kaggle competition," arXiv preprint arXiv:1706.06169, 2017.
[34] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin et al., "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," arXiv preprint arXiv:1603.04467, 2016.
[35] S. Sharma, S. Sharma, and A. Athaiya, "Activation functions in neural networks," Towards Data Sci, vol. 6, no. 12, pp. 310–316, 2017.
[36] B. Ding, H. Qian, and J. Zhou, "Activation functions and their character- istics in deep neural networks," in 2018 Chinese Control And Decision Conference (CCDC), 2018, pp. 1836–1841.
[37] J. Brownlee, "What is the difference between a batch and an epoch in a neural network," Machine Learning Mastery, vol. 20, 2018.
[38] J. T. Barron, "A general and adaptive robust loss function," in Pro- ceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4331–4339.

[39] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.
[40] A. Lydia and S. Francis, “Adagrad—an optimizer for stochastic gradient descent,” Int. J. Inf. Comput. Sci, vol. 6, no. 5, pp. 566–568, 2019.
[41] A. M. Taqi, A. Awad, F. Al-Azzo, and M. Milanova, “The impact of multi-optimizers and data augmentation on tensorflow convolutional neural network performance,” in 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), 2018, pp. 140–145.
[42] P. Gulati, A. Sharma, and M. Gupta, “Theoretical study of decision tree algorithms to identify pivotal factors for performance improvement: A review,” Int. J. Comput. Appl, vol. 141, no. 14, pp. 19–25, 2016.
[43] J. Davis and M. Goadrich, “The relationship between precision-recall and roc curves,” in Proceedings of the 23rd international conference on Machine learning, 2006, pp. 233–240.
[44] T. B. Pathak, M. L. Maskey, J. A. Dahlberg, F. Kearns, K. M. Bali, and D. Zaccaria, “Climate change trends and impacts on california agriculture: a detailed review,” Agronomy, vol. 8, no. 3, p. 25, 2018.
[45] N. Mancosu, R. L. Snyder, G. Kyriakakis, and D. Spano, “Water scarcity and future challenges for food production,” Water, vol. 7, no. 3, pp. 975– 992, 2015.
[46] PlantDoc Dataset Documentation (2023). Dataset Summary for Visual Plant Disease Detection. DatasetNinja repository
[47] Sharma et al. (2024). "Prediction of best suitable crop using machine learning technique." International Journal of Intelligent Systems and Applications in Engineering, 12(23s), 2743
[48] IJISAE (2024). "Prediction of best suitable crop using machine learning technique: Ensemble stacking methodology"
[49] Yadav et al. (2025). "Time series forecasting of crop yield under climate variability using IoT data and LSTM networks." International Journal of Environmental Sciences
[50] Peeriga et al. (2024). "Real-time rain prediction in agriculture using AI and IoT: A bi-directional LSTM approach." Engineering, Technology & Applied Science Research, 14(4), 15805-15812
[51] ADaSci (2024). "Implementing rapid LLM inferencing using Groq." Retrieved from https://adasci.org
[52] DataCamp (2024). "Groq LPU inference engine tutorial." LLamaIndex framework documentation
[53] GeeksforGeeks (2025). "Beginner's guide to Groq API with Llama 3." NLP and AI tutorials
[54] OpenWeather (2025). "Weather data for agriculture -- Forecasts & crop insights." OpenWeather Agriculture Solutions
[55] OpenWeather API Documentation (2024). Historical weather data and forecasting services
[56] Singh et al. (2020). "PlantDoc: A dataset for visual plant disease detection." ACM India Joint International Conference on Data Science and Management of Data (CoDS-COMAD 2020)
[57] He et al. (2015). "Deep Residual Learning for Image Recognition." arXiv:1512.03385



10.22214/IJRASET

INDEX COPERNICUS 45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY