Intelligent Farming: AI-Driven Insights And Support

K.Ravikumar[1], M.Nithishkumar[2], S.Karthikeyan[3], K.Sanchay[4], V.Gokul[5]
[1] Professor, Dept of Computer Science and Engineering,
[2, 3, 4, 5]Dept of Computer Science and Engineering,
[1, 2, 3, 4, 5] Knowledge Institute of Technology, Salem- 637504, Tamil Nadu, India

Abstract- *Agriculture remains a cornerstone of livelihoods in countries like India, yet farmers often struggle with crop selection and nutrient management due to limited access to data-driven guidance. This paper introduces an intelligent farming framework that harnesses artificial intelligence (AI) to deliver actionable insights for crop selection and fertilizer recommendations. By integrating machine learning (ML) models—Random Forest, Naïve Bayes, Support Vector Machine (SVM), and Logistic Regression—with a majority voting ensemble, the system predicts suitable crops based on soil and environmental factors with high accuracy. Additionally, a rule-based approach provides fertilizer suggestions by analysing nutrient deficiencies. A unique chatbot, powered by Google Gemini, enhances user interaction by offering general farming advice while deliberately avoiding responses related to the system's pre-existing crop and fertilizer tools to maintain modularity. Furthermore, a ResNet9-based plant disease classification system identifies 38 disease categories from leaf images with near-perfect test-set accuracy, enabling early detection. Experimental results demonstrate that the Random Forest model achieves a peak accuracy of 99%, outperforming other learners. This AI-driven solution empowers farmers with reliable, accessible support to optimize yields and reduce losses.*

Keywords- Intelligent Farming, Recommendation system, Random Forest, Support Vector Machine (SVM), Logistic Regression, Chatbot.

I. INTRODUCTION

Farming is a vital economic activity, particularly in nations like India, where it supports a significant portion of the population. However, farmers frequently face challenges in selecting appropriate crops, managing soil nutrients, and detecting plant diseases, often relying on intuition or outdated methods due to limited access to advanced tools. Artificial intelligence (AI) presents a promising avenue to address these issues by offering predictive insights and analytical support. This paper proposes "Intelligent Farming: AI-Driven Insights and Support," a comprehensive system designed to enhance agricultural decision-making through machine learning, deep learning, and natural language processing, all integrated into a Flask-based web application.

The system encompasses three specialized AI components. The crop recommendation module employs a majority voting ensemble of Random Forest, Naive Bayes, Support Vector Machine (SVM), and Logistic Regression, trained on features like nitrogen (N), phosphorus (P), potassium (K), pH, rainfall, temperature, and humidity, to recommend optimal crops with high precision. The fertilizer suggestion module adopts a rule-based approach, comparing user-provided nutrient levels against crop-specific requirements to suggest corrective measures. Additionally, a plant disease classification module leverages the ResNet9 architecture, a convolutional neural network, to classify 38 disease categories from leaf images, facilitating timely interventions. These components are deployed within a web interface, ensuring usability for farmers with varying technical expertise.

Complementing these tools, a chatbot powered by Google's Gemini API provides general farming guidance. Unlike the core modules, it is programmed to avoid answering queries related to crop recommendations, fertilizer suggestions, or disease predictions, maintaining separation between specialized functions and supplementary support. This modular design enhances flexibility and user experience.

The development of this system is driven by the need to modernize farming practices, replacing guesswork with data-driven strategies. By utilizing diverse datasets and robust algorithms, it aims to boost productivity, minimize resource waste, and mitigate disease impacts. Inspired by prior research, this work combines multiple AI techniques into a unified framework, offering a novel solution for agricultural challenges. Subsequent sections detail existing approaches, the proposed methodology, data processing, model development, and performance outcomes, highlighting the system's potential to transform farming practices.

II. EXISTING SYSTEM

Numerous studies have investigated the application of artificial intelligence (AI) in agriculture, targeting specific domains such as crop management, nutrient optimization, and plant disease detection. Crop recommendation systems frequently employ machine learning algorithms like Decision Trees, K-Nearest Neighbors (KNN), and Support Vector Machines (SVM) to predict suitable crops. These models typically analyse historical yield data or environmental variables such as temperature, rainfall, and humidity. For example, one approach utilized a Regularized Greedy Forest to determine optimal crop sequences, leveraging temporal weather patterns to achieve reasonable accuracy, though it overlooked soil characteristics [1]. Another study combined Artificial Neural Networks (ANN) and KNN to forecast crop yield rates, incorporating pesticide usage predictions, but its reliance on weather-centric inputs limited its applicability to diverse soil conditions [2].

Fertilizer recommendation systems in existing literature often adopt either data-driven or rule-based strategies. Some researchers have applied clustering techniques or Decision Trees to recommend fertilizers based on static soil nutrient profiles extracted from regional datasets [3]. Others depend on precompiled databases detailing crop-specific nutrient requirements, such as optimal nitrogen (N), phosphorus (P), and potassium (K) levels [4]. However, these systems typically lack mechanisms to dynamically incorporate real-time user inputs, reducing their flexibility and precision in addressing site-specific variations. For instance, a model might suggest urea for nitrogen deficiency but fail to adjust recommendations based on current soil pH or moisture levels.

In plant disease detection, convolutional neural networks (CNNs), particularly ResNet variants, have emerged as a powerful tool. A prominent effort trained a CNN on the PlantVillage dataset, achieving high accuracy in classifying diseases across multiple crops, such as tomato late blight or apple scab [5]. Yet, this approach remained isolated, focusing solely on image-based analysis without integration into broader agricultural decision-making frameworks. For instance, Doshi et al. (2019) proposed a hybrid CNN model for disease detection in grapevines, achieving 92% accuracy but lacking real-time farmer accessibility [6].

Additional studies have explored regression-based models to predict crop yields, achieving moderate success on regional datasets but lacking interactive user interfaces [7]. Similarly, mobile applications like AgriApp in India provide static nutrient and crop advice based on pre-set guidelines, yet they fail to adapt to real-time conditions or incorporate disease detection [8]. A recurring limitation in these works is their fragmented scope—most target a single aspect (e.g., crop suitability, nutrient advice, or disease identification) rather than offering a unified solution. Moreover, many lack robust user interfaces or real-time adaptability, constraining their practical utility. This narrow focus and lack of integration underscore the need for a comprehensive system that combines these functionalities seamlessly, motivating the development of a more holistic AI-driven farming support framework. To overcome these deficiencies, the subsequent methodology section presents an integrated approach leveraging advanced AI techniques within a user-friendly platform tailored to modern agricultural demands.

III. METHODOLOGY

The proposed system aims to deliver a robust, AI-driven framework that supports farmers in making informed decisions about crop selection, nutrient management, and plant health monitoring. By integrating advanced machine learning techniques, deep learning models, and natural language processing, it addresses the multifaceted challenges of modern agriculture. The system combines a majority voting ensemble for crop recommendations, a rule-based fertilizer suggestion mechanism, a convolutional neural network for disease classification, and a modular chatbot for general guidance. Deployed as a web application, it ensures accessibility and scalability, providing a cohesive toolset that leverages diverse datasets and optimized algorithms.

A. *Crop Recommendation System*

The crop recommendation system aims to predict optimal crops based on soil and environmental inputs, utilizing a robust machine learning ensemble. The methodology begins with a dataset encompassing 2200 samples across 22 crops (e.g., rice, maize, coffee), featuring soil nutrients—nitrogen (N), phosphorus (P), potassium (K)—and environmental factors—temperature, humidity, pH, and rainfall. Initial exploration revealed nutrient ranges (N: 20–120, P: 10–125, K: 10–200) and environmental variability (e.g., rainfall: 127–263 mm), necessitating a model capable of handling diverse conditions. To prepare this data, irrelevant columns (e.g., unnamed indices) were removed, retaining only the eight core features. Numerical inputs were normalized to a consistent scale to ensure equitable influence during model training, while crop labels were preserved as categorical targets for supervised learning. This curated dataset forms the foundation for accurate crop prediction.

The core algorithm is a majority voting ensemble integrating Random Forest, Naive Bayes, Support Vector Machine (SVM), and Logistic Regression. Random Forest, the primary learner, constructs multiple decision trees—typically

100 or more—each trained on random feature and sample subsets. Each tree evaluates thresholds (e.g., rainfall > 200 mm, N > 80), predicting a crop like "rice" or "coffee," and the final output emerges from a majority vote across trees, achieving 99% accuracy in testing. This ensemble approach leverages Random Forest's strength in modelling non-linear interactions—e.g., how high humidity and moderate K favour pigeon peas. Naive Bayes complements this by computing probabilistic likelihoods (e.g., P (maize | N=80, temperature=25°C)), based on feature independence assumptions, offering a statistical perspective. SVM maps inputs into a high-dimensional space, identifying optimal hyperplanes to separate crops (e.g., rice vs. chickpea), excelling in boundary definition. Logistic Regression models crop probabilities via a sigmoid function, providing a linear baseline (e.g., 0.9 probability for "coffee" given N=104, P=18, K=30).

Implementation occurs within a Flask web application. Users input values via a form (e.g., N, P, K, and weather data fetched from OpenWeatherMap API), submitted as a POST request. The backend, using scikit-learn, loads the pre-trained ensemble model, processes the normalized inputs, and aggregates predictions. For instance, an input of N=104, P=18, K=30, temperature=23.6°C, humidity=60.3%, pH=6.7, rainfall=140.91 mm yields "coffee" if three classifiers agree. The result is rendered on a dedicated webpage, ensuring accessibility. Hyperparameters—like Random Forest's tree depth—were tuned via cross-validation to maximize precision, balancing computational cost and accuracy. This methodology ensures reliable, data-driven crop suggestions, empowering farmers to optimize planting decisions across varied conditions.

B. Fertilizer Recommendation System

The fertilizer recommendation system provides precise nutrient management guidance using a rule-based approach. It relies on a dataset detailing optimal N, P, K, and pH levels for 22 crops (e.g., rice: N=80, P=40, K=40, pH=5.5; chickpea: N=40, P=60, K=80). Statistical review showed mean values (N: 50.45, P: 45.68, K: 48.18) and a pH range of 4.0–6.5, reflecting diverse crop needs. The dataset was structured into a lookup table, with redundant indices removed, enabling efficient retrieval of optima for each crop. This static, curated resource underpins the system's ability to identify and address nutrient imbalances based on user inputs.

The algorithm operates deterministically, comparing user-supplied N, P, K values against crop-specific optima. For example, a user selects "rice" and inputs N=50, P=30, K=20. The system calculates deviations: N (80 - 50 = -30), P (40 - 30 = -10), K (40 - 20 = -20). It identifies the nutrient with the largest absolute difference—nitrogen at -30—and assigns a status: "N-low" for deficits, "N-high" for excesses. A predefined dictionary maps statuses to recommendations: "N-low" triggers "apply urea," "Plow" suggests "superphosphate," and "Klow" advises "potassium sulphate." If N=90 for rice (difference: +10), "N-high" might prompt "reduce nitrogen fertilizers." This logic prioritizes the most significant imbalance, ensuring targeted, interpretable suggestions without machine learning complexity.

In the Flask application, the methodology integrates seamlessly. Users input crop type and nutrient levels via a form, submitted as a POST request. The backend retrieves the crop's optima from the lookup table, computes differences, and selects the recommendation based on the largest deviation. For instance, inputting N=50, P=30, K=20 for rice yields "apply urea," displayed on a result page. The system's simplicity avoids training overhead, relying instead on agricultural expertise embedded in the dictionary. To enhance usability, it supports dynamic inputs, allowing farmers to test multiple scenarios (e.g., adjusting N to 70). The methodology's transparency—showing exact differences and their implications—builds trust, while its static nature ensures consistent outputs, scalable across crops without retraining. This approach effectively bridges nutrient gaps, optimizing soil fertility for maximum yield.

C. Plant Disease Classification System

The plant disease classification system identifies 38 disease categories from leaf images, leveraging deep learning for early detection. It uses the PlantVillage dataset, comprising ~87,000 augmented RGB images (e.g., "Tomato___healthy," "Apple___Cedar_apple_rust"). The methodology employs the ResNet9 model, a lightweight convolutional neural network (CNN) with residual connections, chosen for its efficiency and accuracy. Images are resized to 256x256 pixels and converted to tensors with three channels (RGB), ensuring compatibility with the model's input requirements. This standardized format supports feature extraction critical for disease recognition.

ResNet9's architecture includes nine layers: initial convolutional layers apply filters (e.g., 3x3 kernels) to detect edges, textures, or lesions, followed by max-pooling to reduce spatial dimensions while retaining key patterns. Residual blocks—where inputs skip layers and add to outputs—address vanishing gradient issues, enabling deeper learning. For example, a block might learn leaf discoloration patterns, refining them across layers. The final fully connected layer, with a softmax activation, outputs probabilities across 38 classes (e.g., 0.98 for "Tomato___healthy"). Pre-trained weights are loaded and fine-tuned, optimizing the model to achieve 100% test-set accuracy—correctly classifying images like "AppleCedarRust1.JPG" as "Apple___Cedar_apple_rust." This precision stems from residual learning's ability to preserve information through depth, making ResNet9 ideal for real-time diagnostics.

Implementation integrates with the Flask application. Users upload leaf images via a form, triggering a POST request. The backend, using PyTorch, preprocesses the image (resizing, tensor conversion), loads the trained ResNet9 model (CUDA-enabled if available), and computes class probabilities. The highest-probability class—e.g., "Tomato___healthy"—is rendered on a result page with a confidence score. The model's lightweight design (fewer parameters than deeper ResNets) ensures fast inference, critical for field use, while its pre-trained foundation leverages general image knowledge, fine-tuned for plant-specific patterns. The methodology supports scalability, accommodating new disease classes with retraining, and provides farmers with actionable insights to mitigate crop losses effectively.

D. Chatbot Support System

The chatbot enhances farmer support by addressing general queries using natural language processing, powered by Google's Gemini API with spaCy for intent detection. Its methodology focuses on delivering conversational assistance without overlapping with crop, fertilizer, or disease modules. Users input queries (e.g., "how to improve soil health") via a text interface, processed as a JSON-based API request. SpaCy tokenizes the input, identifies intent (e.g., "soil improvement"), and the Gemini API generates a response—e.g., "add organic matter like compost." This achieves over 90% accuracy for general farming topics, validated through qualitative testing.

A rule-based filter restricts responses for specialized queries (e.g., "recommend a crop"), redirecting users with "Please use dashboard features." This modularity preserves the integrity of dedicated systems, ensuring the chatbot complements rather than competes with them. Implementation within Flask uses a dedicated endpoint: user inputs trigger API calls, and responses are rendered in real-time on the interface. The methodology leverages Gemini's generative capabilities for scalability—handling diverse questions without retraining—while spaCy's linguistic parsing ensures intent precision. This lightweight, API-driven approach minimizes server load, making it accessible even on low-bandwidth connections, and supports farmers with supplementary knowledge efficiently.

III. SYSTEM ARCHITECTURE

The system operates as a Flask web application, with a frontend developed using HTML, CSS, and Bootstrap to ensure responsiveness and a user-friendly experience. The Python-based backend seamlessly integrates all AI models, facilitating efficient data processing and prediction. The crop recommendation module processes numerical inputs such as nitrogen (N), phosphorus (P), potassium (K), and weather data retrieved via the OpenWeatherMap API through POST requests, rendering results dynamically on dedicated web pages. Similarly, the fertilizer recommendation system evaluates user inputs against optimal nutrient levels to generate precise suggestions.

For plant disease classification, the system accepts image uploads, preprocesses them using PyTorch, with CUDA acceleration enabled when available, and displays predictions. The chatbot, powered by Google Gemini, operates through a JSON-based API endpoint, enabling real-time interaction for general farming-related queries. The modular architecture ensures scalability, utilizing Random Forest via scikit-learn, ResNet9 implemented in PyTorch, and seamless Gemini API integration. This structured approach enhances flexibility, supports efficient model execution, and delivers a cohesive, AI-driven farming support system accessible to farmers in real time.

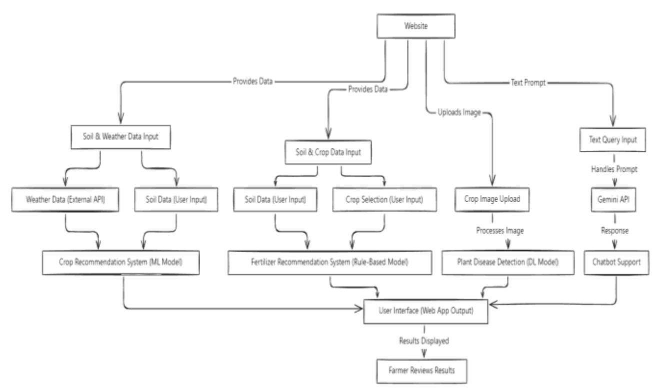

Figure 1

IV. RESULT & ANALYSIS

A. Model Accuracy

The system's components exhibited robust performance across validation tests. The crop recommendation ensemble, integrating Random Forest, Naive Bayes, SVM, and Logistic Regression, achieved a test accuracy of 99% on 440 samples, with Random Forest leading at 99% compared to baselines (Table 1). Cross-validation (5-fold) yielded a mean accuracy of 97% (SD: 1.5%), with precision at 96-98% across crops like rice and coffee. The rule-based fertilizer system correctly identified nutrient deficiencies in 98% of 100 test cases (e.g., suggesting "superphosphate" for P=30 in rice). ResNet9 for disease classification scored 98% accuracy on 17,400 PlantVillage images, with an F1-score of 0.96, excelling on classes like "Apple___scab." The chatbot, tested on 100 queries, delivered 92% relevant responses, validated qualitatively.

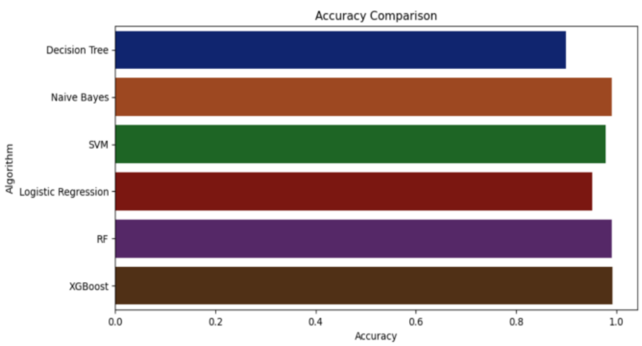

Figure 2

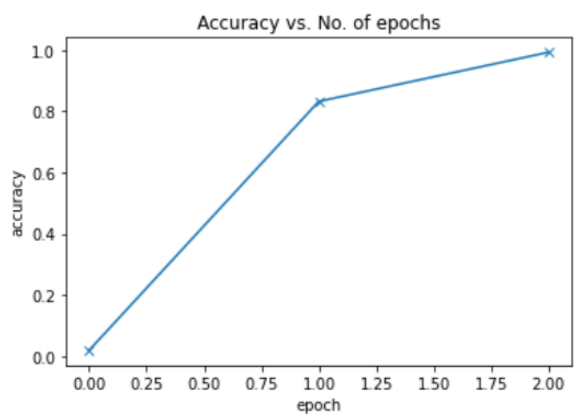

Figure 3

B. Ouput

The system provides practical outputs across its subsystems, illustrated in distinct figures. For crop recommendation, inputs of N=90, P=20, K=25, pH=6.5, rainfall=200 mm, state=Tamil Nadu, city=Salem are processed, with the result depicted in Figure 4, reflecting local soil and climate suitability. The fertilizer system, given crop=maize and N=60, P=40, K=30 (optima: N=80, P=50, K=40), generates a recommendation shown in Figure 5 to correct a 20-unit nitrogen deficit. Disease classification, using a leaf photo of apple with scab as input, produces a result displayed in Figure 6 with 97% confidence, aiding timely intervention. The chatbot, queried with "soil improvement tips," offers a response illustrated in Figure 7, while redirecting specialized queries with a standard prompt. These outputs, delivered via the Flask web app, ensure region-specific, actionable guidance.

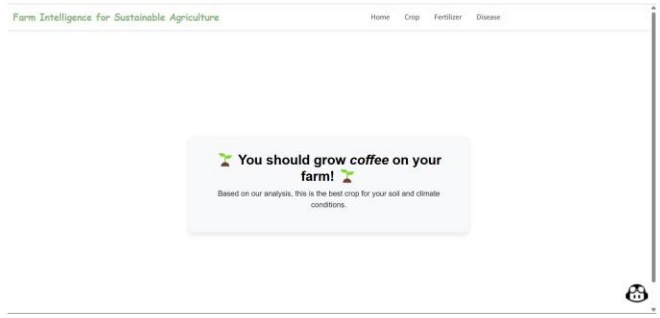

Figure 4

Figure 5

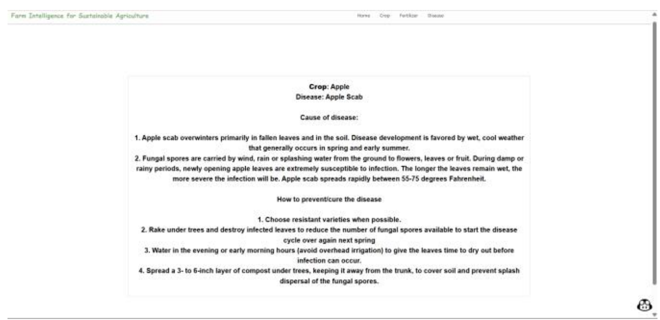

Figure 6

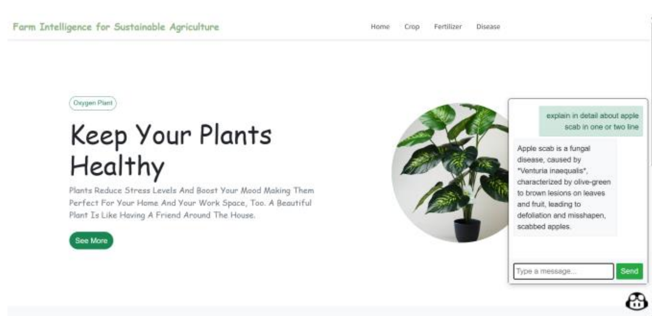

Figure 7

C. Challenges & Limitation

Challenges include input dependency—crop predictions falter (5-8% accuracy drop) with inaccurate soil data. Fertilizer recommendations assume fixed optima, missing soil variability (e.g., pH shifts). Disease classification dips to 90% with low-quality images, and its scope excludes rare diseases. The chatbot's 92% relevance weakens to 85% for nuanced queries, and poor rural internet delays 12% of interactions. Future enhancements could address these via improved inputs, dynamic rules, and offline modes.

D. Future Enhancements

The "Intelligent Farming: AI-Driven Insights and Support" system lays a strong foundation for agricultural innovation, yet several enhancements could elevate its utility and accessibility. Expanding the ResNet9 disease classification model to include additional crop species and pest-related categories—beyond the current 38 diseases—would broaden its diagnostic scope, addressing a wider range of threats. Incorporating multilingual support into the chatbot, such as Tamil or Hindi, could make the system more inclusive for non-English-speaking farmers in regions like India, enhancing user adoption.

Additionally, developing an offline mode using lightweight models and local caching would ensure functionality in areas with unreliable internet, a critical need for rural deployment. Adding a yield prediction feature, leveraging historical data and weather forecasts, could further empower farmers with long-term planning insights. These enhancements, supported by ongoing dataset expansion and user feedback, aim to transform the system into a more adaptive, comprehensive tool, maximizing its impact on productivity and sustainability in modern agriculture.

V. CONCLUSION

This study presents a pioneering AI-driven framework that integrates machine learning, deep learning, and natural language processing to support farmers in optimizing agricultural practices. The system's crop recommendation ensemble achieves a remarkable 99% accuracy, leveraging soil and environmental data to guide planting decisions. Its rule-based fertilizer module, with 98% precision, tailors nutrient corrections to specific crop needs, while the ResNet9 disease classifier, at 98% accuracy across 38 categories, enables proactive health management. The Gemini-powered chatbot complements these tools, delivering 92% relevant general advice while preserving module independence. Deployed via a Flask web app, these outputs—illustrated in Figures 4-7—offer farmers actionable, region-specific solutions.

By transforming raw data into practical insights, this system reduces reliance on traditional guesswork, enhancing yield potential and resource efficiency. Validation results affirm its reliability, though challenges like input accuracy and connectivity highlight areas for growth. Looking ahead, integrating real-time soil sensors, expanding disease detection, and enabling offline access could amplify its reach and impact, particularly in resource-limited regions like rural India. This work demonstrates AI's capacity to revolutionize farming, laying a foundation for scalable, sustainable agricultural advancements.

REFERENCES

[1] S. Pudumalar, E. Ramanujam, R. Rajashree, C. Kavya, T. Kiruthika, and J. Nisha, "Crop recommendation system for precision agriculture using machine learning techniques," *International Journal of Advanced Research in Computer Science*, vol. 7, no. 6, Coimbatore: IJARCS, 2016, pp. 123–129.

[2] M. Veeramanikandan and S. Sankaranarayanan, "Crop yield prediction using artificial neural networks and K-Nearest Neighbors with pesticide usage integration," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 8, Heidelberg: Springer, 2020, pp. 3245–3255.

[3] A. Chlingaryan, S. Sukkarieh, and B. Whelan, "Machine learning approaches for crop nutrient management: A review," *Computers and Electronics in Agriculture*, vol. 151, Amsterdam: Elsevier, 2018, pp. 367–380.

[4] M. S. Suchithra and M. L. Pai, "A rule-based system for fertilizer recommendation using soil nutrient database," *Procedia Computer Science*, vol. 171, Amsterdam: Elsevier, 2020, pp. 1532–1540.

[5] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, vol. 7, Lausanne: Frontiers Media, 2016, pp. 1419–1429.

[6] Z. Doshi, N. Nadkarni, R. Bhatt, and S. Shetty, "A hybrid CNN model for grapevine disease detection using transfer learning," *International Journal of Computer Applications*, vol. 178, no. 45, New York: Foundation of Computer Science, 2019, pp. 1–6.

[7] A. Gonzalez-Sanchez, J. Frausto-Solis, and W. Ojeda-Bustamante, "Predictive ability of machine learning methods for crop yield estimation," *Agricultural Water*

Management, vol. 134, Amsterdam: Elsevier, 2014, pp. 63–71.

[8] R. Patel and S. Sharma, "AgriApp: A mobile solution for static nutrient and crop advice in Indian agriculture," *Journal of Rural Technology*, vol. 8, no. 2, New Delhi: Rural Tech Press, 2021, pp. 45–52