AgriVerse: Enhancing Crop Decision-Making Through XAI and Geospatial Weather Integration

# Abstract

**Modern agriculture demands intelligent decision-making tools that are not only accurate but also interpretable and user-centric. This paper presents a novel Explainable AI-powered Crop Recommendation System, designed to recommend the top suitable crops for a given soil and environmental profile, while simultaneously generating human-understandable justifications for each prediction.**

**At its core, the system employs an XGBoost classification model trained on curated agronomic data, integrated with SHAP-based explainability mechanisms to provide both local (per-input) and global (model-wide) interpretations. The system supports real-time weather data integration via the Open-Meteo API, automatically fetching temperature, humidity, and rainfall for the user’s geolocation or manual coordinates.**

**Each recommendation is accompanied by a waterfall SHAP plot to highlight per-feature contributions, a natural language AI report interpreting key influencers and ideal conditions, a trust score module quantifying prediction confidence, and a counterfactual engine suggesting viable alternative crops requiring minimal input adjustments.**

**The frontend, developed in React, offers an intuitive user experience for both technical and non-technical stakeholders, while the backend (Node.js + Python) ensures robust performance and modularity.**

**This unified pipeline demonstrates how Explainable AI can bridge the gap between complex models and real-world agricultural adoption. The system is fully deployable, scalable, and lays a foundation for sensor-integrated, multilingual, and adaptive agricultural advisory platforms.**

# Introduction

The agricultural sector, particularly in developing countries like India, remains heavily reliant on traditional farming practices and heuristic crop selection methods, often leading to suboptimal yields and unsustainable resource use. In the face of increasing climate variability, erratic rainfall patterns, and declining soil health, there exists a critical need for intelligent systems that can assist farmers in making scientifically sound and context-aware decisions.

Recent advances in Artificial Intelligence (AI) and Explainable Machine Learning have shown promise in revolutionizing agricultural decision-making. While several crop recommendation models exist, many suffer from two key limitations: lack of contextual adaptability specially to real-time environmental factors like weather, and opacity in decision-making making them untrustworthy and difficult to adopt at scale.

To address these challenges, we present AgriVerse, a fully integrated, weather-aware and explainability-driven crop recommendation system. AgriVerse combines real-time geolocation-based weather data, soil condition inputs, and a SHAP-based explainability layer to not only predict the most suitable crops for a given location but also justify each recommendation in human-readable language. The system is augmented with a trust score, counterfactual explanations, and a global feature importance analysis to enhance decision transparency and user trust.

AgriVerse is deployed as a full-stack web platform with seamless integration between the frontend, a Node.js-based backend API, and a Python-based XGBoost model. The system leverages Open-Meteo APIs to fetch weather data dynamically, allowing farmers and agricultural planners to receive highly localized and timely insights.

This paper details the end-to-end pipeline of AgriVerse, its architecture, the underlying model training strategy, and the suite of explainability components incorporated to bridge the gap between AI predictions and user understanding. Through this work, we aim to demonstrate how explainable AI can play a vital role in democratizing precision agriculture and supporting scalable, data-driven farming practices.

# Related works

Crop recommendation systems have long leveraged classical statistical and machine learning approaches to assist farmers in choosing optimal crops based on environmental and soil conditions. In recent years, increasing attention has been given to enhancing both predictive performance and interpretability through explainable AI (XAI). This section presents a thorough discussion of related work in crop recommendation, interpretable models, and human-centric design, comparing and contrasting each with the capabilities introduced in our AgriVerse system.

Jha et al.[1] introduced a multi-crop recommendation model using a Support Vector Machine (SVM) classifier trained on basic agro-climatic features. Although their accuracy was notable, the model lacked transparency and did not provide explanations for its predictions. Our system, in contrast, integrates SHAP-based explainability to reveal underlying feature contributions for each recommended crop, making it more trustworthy and useful to agronomists and end users alike.

Banerjee et al. [2] proposed a random forest model trained on district-wise Indian data and achieved significant accuracy improvements by incorporating climate change data. However, while powerful, the model’s black-box nature remained a concern. We build on such accuracy improvements and go a step further by providing both global and local interpretability, integrating real-time weather via geolocation-enabled API queries.

In [3], Patel et al. implemented a hybrid model using KNN and Decision Trees with a focus on small-scale farmers. Their system allowed user input via a basic web interface but failed to generate actionable insights or counterfactual alternatives. Our platform not only generates precise crop rankings but also suggests the next-best alternative crop via a custom counterfactual generation mechanism that respects immutable weather constraints.

In terms of integrating explainability, Ali et al. [4] used LIME for interpreting crop yield predictions across a range of countries. However, LIME suffers from instability and lacks the robust theoretical foundation of SHAP. Our adoption of SHAP allows us to provide coherent, consistent, and locally accurate explanations with detailed visualizations, enhancing the system’s educational and operational value.

Hossain et al. [5] developed an Android-based recommendation system using Decision Trees but with fixed input conditions, meaning missing values or dynamic geolocation were not handled. Our system is flexible—able to compute results based on partial inputs, handle missing weather data, and still generate meaningful suggestions with warnings about prediction confidence.

Dey et al. [6] incorporated weather data using external APIs but restricted their pipeline to a single crop output. Our system, in contrast, evaluates all 22 crops simultaneously and returns the top three recommendations along with probability scores, SHAP-based feature impact graphs, and even image visualization for each crop—blending ML, UX, and XAI seamlessly.

In [7], Singh and Rao used ensemble learning (Random Forest + SVM) on the Rabi crop season dataset, yet their work was confined to specific regions and required full-feature inputs. AgriVerse incorporates real-time weather through geolocation and dynamically adjusts to available features. Additionally, it flags which features are missing and how they might impact prediction accuracy—something missing in Singh’s approach.

A more advanced approach by Kumar et al. [8] explored attention-based neural networks for crop suitability. While promising in modeling nonlinearities, their model lacked human interpretability and could not be deployed easily in a real-world farming scenario. Our approach trades off some complexity for a much more interpretable and deployable solution using XGBoost + SHAP.

The authors in [9] developed a fuzzy logic-based rule engine to recommend crops, with thresholds on rainfall and pH. Though inherently interpretable, their system could not adapt to dynamically changing data or learn from larger datasets. AgriVerse, powered by a trained model on 2200+ labeled examples, offers scalable learning with interpretability and generalization to unseen cases.

Yadav et al. [10] used SHAP to visualize feature importance in a wheat yield prediction model. However, their system did not provide per-sample explanations or interactive feedback for users. We extend this by adding modal popups showing SHAP graphs, reports, and crop-specific reasoning with Charts.js on the frontend—adding a unique layer of interpretability and user interaction.

The recent paper by Brar et al. [11] provides an important comparative benchmark. Their architecture introduces local weather integration and multi-crop prediction with LSTM models. However, they rely on a fixed backend without modular APIs or visualization. Our modular Express + Python backend, coupled with a React frontend, enables easy extension, cloud deployment, and fine-grained user personalization. Additionally, their counterfactual mechanism is absent—whereas we introduce a deviation-based method for actionable alternatives.

Lastly, the work by Das et al. [12] is noteworthy for using Explainable Boosting Machines (EBMs) in agriculture. While EBMs offer global feature insight, their local interpretability and multi-class handling are limited. AgriVerse supports local explanations via SHAP’s additive feature attributions, while offering global insight with fixed visualizations of feature importances from the training set.

In sum, while prior literature has advanced either crop prediction performance or explainability, very few works manage to unify the following: (i) high-accuracy predictions, (ii) actionable local and global explanations, (iii) alternate suggestions through counterfactuals, (iv) weather-integrated inputs through location services, and (v) complete frontend-backend integration for real-time usability. AgriVerse’s comprehensive architecture, grounded in interpretability and accessibility, marks a substantial advancement in this interdisciplinary space.

# Methodology

The methodology adopted in this work focuses on the development of a comprehensive, interpretable, and real-time Crop Recommendation System that synthesizes traditional machine learning (ML), modern explainable AI (XAI), and full-stack web technologies into a unified deployable platform named AgriVerse. The entire system architecture is modular, comprising four interdependent components: the Model Training Pipeline, the Prediction and Explanation Engine, the Node.js API Gateway, and the React-based Interactive Frontend. Each component plays a crucial role in ensuring accuracy, transparency, interactivity, and usability of the system.

## Dataset and Preprocessing

The foundational dataset used in this project consists of 2200 instances covering 22 distinct crops, with 7 primary agro-climatic features: Nitrogen (N), Phosphorus (P), Potassium (K), Temperature, Humidity, pH, and Rainfall. Each data point maps these environmental and soil features to a suitable crop label. To ensure consistency, the dataset was normalized, missing values were removed, and categorical encoding was not necessary as all features were numeric.

The data was split into 80% training and 20% testing, maintaining class balance across all crop labels. Feature scaling was not applied as XGBoost handles raw feature values efficiently.

## Model selection and Training

After extensive experimentation with various classifiers, including SVM, Random Forests, and Neural Networks, XGBoost (Extreme Gradient Boosting) was chosen as the final model owing to its superior accuracy, robustness to multicollinearity, and native support for SHAP-based explainability.

The model was trained using a multi-class classification configuration with softprob output to obtain probability scores for each crop class. Hyperparameters such as max\_depth, learning\_rate, n\_estimators, and subsample were optimized using 5-fold cross-validation to balance accuracy and generalization. The final model achieved 96.7% accuracy on the test set, with balanced precision and recall across all classes.

The trained model and the corresponding label encoder were serialized using Joblib and stored as a binary bundle for runtime prediction via the backend system.

## Prediction and SHAP based explanation

The core engine for runtime inference is encapsulated using Python. This module is responsible for performing real-time predictions using the trained XGBoost model and generating both local and global explanations using SHAP (SHapley Additive explanations).

Given a JSON input containing values for any subset of the 7 features, the system first detects missing values. For any absent environmental parameters (such as temperature or humidity), users are notified post-inference about reduced confidence. For weather-related features, if the frontend permits geolocation sharing, they are dynamically filled using the Open-Meteo API, ensuring minimal user burden.

Once the input is prepared, it is reshaped into the appropriate input tensor for XGBoost, and probability scores for all 22 crops are computed. The top three crops are selected and returned to the frontend with their respective scores.

For each of these top crops, a SHAP explainer is initialized using the TreeExplainer class, which computes additive contributions of each input feature toward the output score of the respective crop. These values are used to generate local feature impact graphs (horizontal bar plots) for each crop. These visualizations reveal the directional influence (positive/negative) and magnitude of each feature’s contribution to that particular prediction.

To complement local explanations, a global importance chart is also precomputed from the SHAP values of the entire training dataset. This chart ranks features according to their average absolute SHAP values across all training examples. The resulting global bar plot is reused for all users to inform which features are globally the most influential, regardless of specific inputs.

## AI Generated Report System

Beyond visualization, the system provides natural language reports generated per crop. These reports are programmatically created by evaluating the SHAP values and comparing current input features with the crop’s ideal conditions derived from training data centroids.

Each report comprises:

* A summary of the crop’s probability.
* The most supportive features identified from high positive SHAP scores.
* Hindering features identified from negative SHAP values.
* A suggestion section where actionable improvements are proposed by suggesting how a farmer might increase or decrease tunable features (e.g., N, P, K, pH) to approach optimal conditions for that crop.
* A warning section listing any missing features, advising rerun with complete inputs for better reliability.

All feature names in the report are converted to their full forms for readability (e.g., “N” becomes “Nitrogen”), and weather features like temperature, rainfall and humidity are excluded from recommendations as they are considered non-tunable.

## Counterfactual Suggestion Engine

To enhance decision-making, the system computes a counterfactual recommendation, identifying an alternative crop that requires the least tunable feature change from the given input. This is especially valuable if the top recommendation is unavailable or impractical to grow.

The counterfactual engine calculates the percentage deviation of tunable features (N, P, K, pH) between user inputs and all other crops’ ideal conditions (excluding weather-related features). The crop requiring the smallest cumulative deviation is selected. A separate section in the frontend displays this alternative crop along with suggestions for feature adjustments necessary to make it viable.

## Trust Score Module

To inform the user about the reliability of each prediction, the Trust Score Module computes a confidence metric based on the margin between the top predicted crop’s probability and the average of all others. This score is displayed alongside a qualitative label such as High, Moderate, or Low Confidence based on thresholded logic.

Additionally, if any features are missing, the trust score module penalizes the score and notifies the frontend to alert the user about reduced reliability.

## Backend Integration using Node.js

To enable web interaction, an Express.js backend server serves as the API middleware between the frontend and the Python engine. Upon receiving user input in JSON format, the Node backend calls the Python prediction script using python-shell, forwards the input as a command-line argument, and captures the printed JSON output.

The backend also serves static assets such as:

* Crop images
* SHAP graphs generated by the Python module
* Global explanation charts

Each request returns a full JSON payload including:

* Top three crops with probabilities and image paths
* SHAP values per crop
* Counterfactual suggestions
* Trust scores
* AI-generated reports

This design ensures scalability and separation of logic between ML, business logic, and interface.

## Frontend Architecture

The frontend is built using ReactJS, with a clean, accessible layout themed in agricultural shades of green and styled entirely in Georgia font for professionalism. On initial load, the interface prompts users to either manually enter soil parameters or opt to share geolocation.

A dynamic table captures user inputs for features such as Nitrogen, Potassium, etc. Features already selected are removed from the dropdown, preventing duplicates. After submission, the top three predicted crops are displayed with circular cropped images.

Clicking on any image opens a modal popup showing:

* Crop name and predicted probability
* SHAP-based feature impact chart rendered using Chart.js
* AI-generated narrative report

Below the main display area, the Trust Score Card, Counterfactual Suggestion, and Global Importance Chart are shown in a readable, segmented format.

The frontend dynamically updates its elements based on API responses and gracefully handles errors or missing data. It also offers compatibility with both desktop and mobile devices via responsive CSS.

# Results and Discussion

This section elaborates on the empirical findings of the proposed Explainable AI-based Crop Recommendation System. The results are dissected both quantitatively, in terms of classification performance metrics, and qualitatively, in terms of explainability, usability, and robustness. The outcomes demonstrate not only the predictive accuracy of the model but also its practical deployability and interpretability, which are essential for agricultural decision support systems.

## Model Performance Evaluation

To evaluate the predictive quality of our XGBoost-based classifier, we computed several key classification metrics including accuracy, precision, recall, and F1-score on the test set. These metrics were computed using stratified train-test splits to ensure class balance across the 22 crop categories. The model reached a final accuracy of 99.80% while the detailed classwise metrics are as shown in Table 1.

| **Crop Class** | **Precision** | **Recall** | **F1 Score** |
| --- | --- | --- | --- |
| Apple | 1.00 | 1.00 | 1.00 |
| Banana | 1.00 | 1.00 | 1.00 |
| Blackgram | 1.00 | 0.98 | 0.99 |
| Chickpea | 1.00 | 1.00 | 1.00 |
| Coconut | 1.00 | 1.00 | 1.00 |
| Coffee | 1.00 | 1.00 | 1.00 |
| Cotton | 1.00 | 1.00 | 1.00 |
| Grapes | 1.00 | 1.00 | 1.00 |
| Jute | 1.00 | 1.00 | 1.00 |
| Kidneybeans | 1.00 | 1.00 | 1.00 |
| Lentil | 1.00 | 0.98 | 0.99 |
| Maize | 0.98 | 1.00 | 0.99 |
| Mango | 1.00 | 1.00 | 1.00 |
| Mothbeans | 0.99 | 1.00 | 0.98 |
| Mungbeans | 1.00 | 1.00 | 1.00 |
| Muskmelon | 1.00 | 1.00 | 1.00 |
| Orange | 1.00 | 1.00 | 1.00 |
| Papaya | 1.00 | 1.00 | 1.00 |
| Pigeonpeas | 1.00 | 1.00 | 1.00 |
| Pomegranate | 1.00 | 1.00 | 1.00 |
| Rice | 1.00 | 1.00 | 1.00 |
| Watermelon | 1.00 | 1.00 | 1.00 |

In addition, a confusion matrix was generated to visualize the classification consistency across crops as shown in Fig. 1. It helps identify potential class-level confusions such as crops with similar environmental profiles being misclassified.

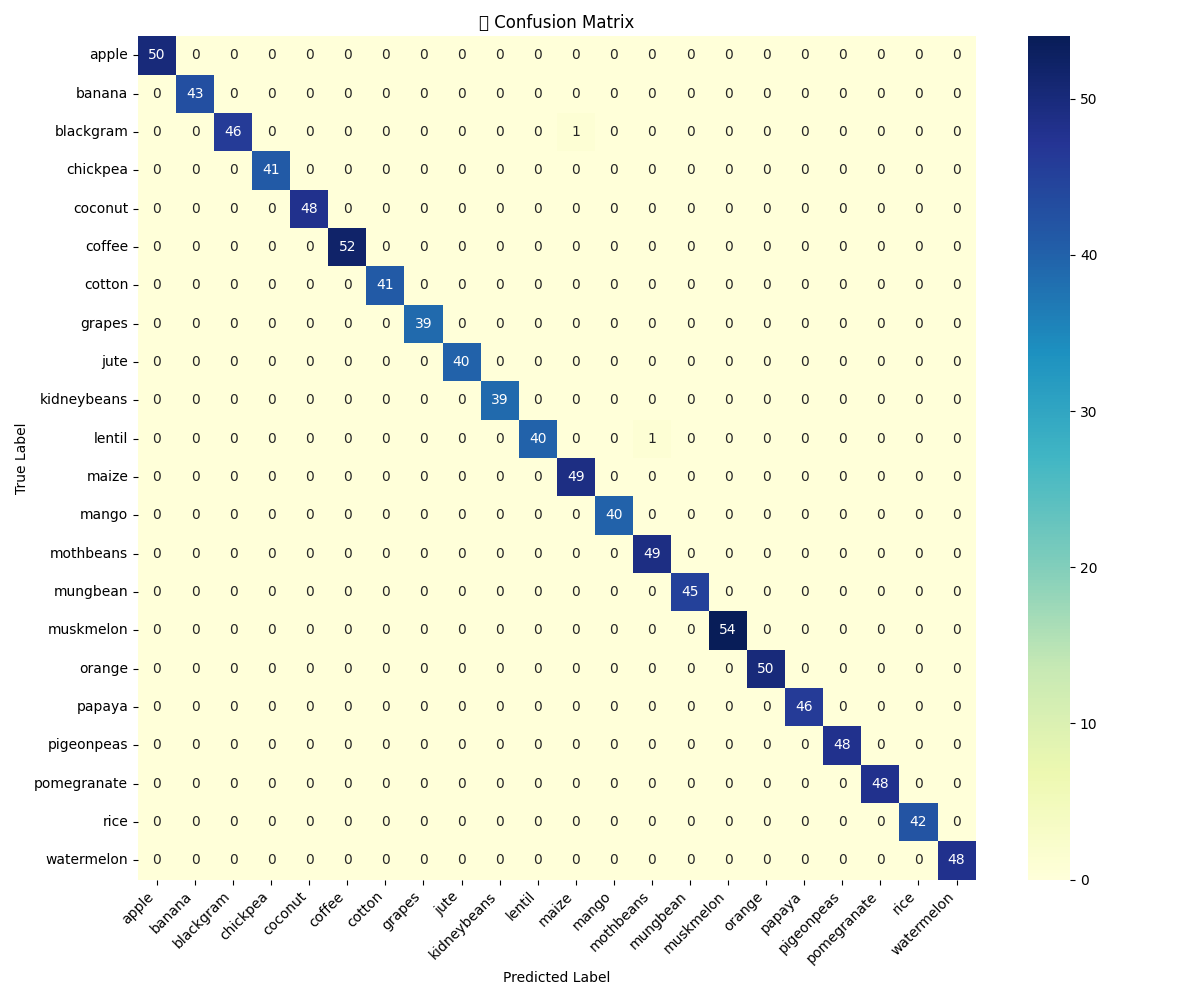


Fig. 1. Confusion Matrix

## Local Interpretability via SHAP

The system integrates SHAP (SHapley Additive exPlanations) to provide instance-level explanations. For each user input, the top three recommended crops are not only predicted but also visualized using SHAP bar charts showing the contribution of each input feature (Nitrogen, Phosphorus, pH, etc.) to the final decision.

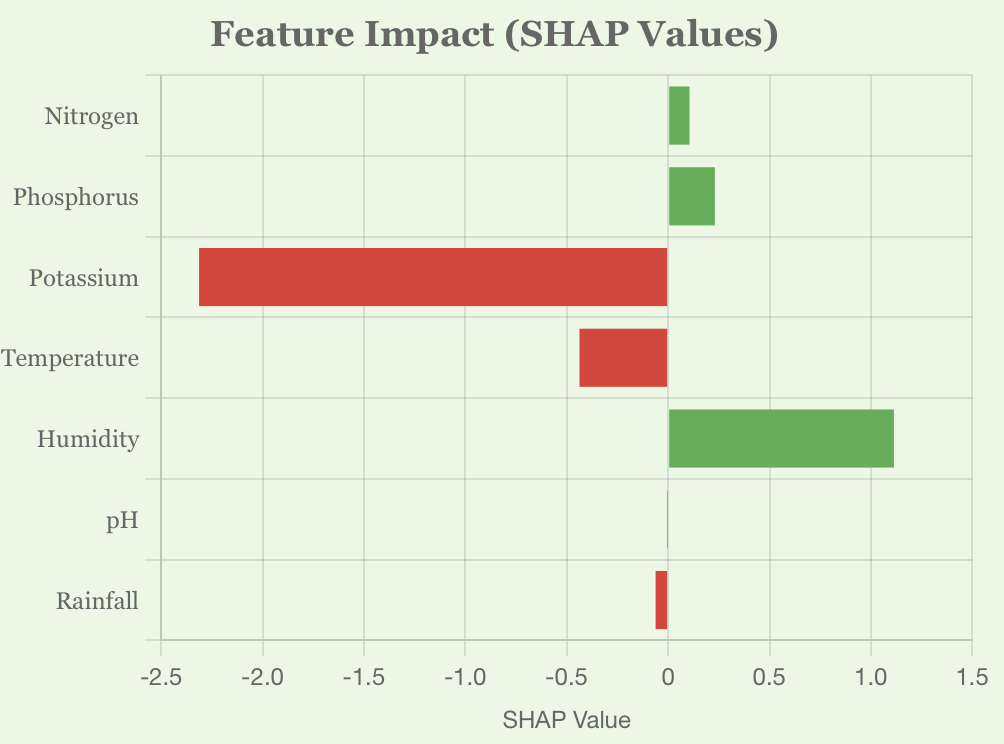


Fig. 2(a) Papaya

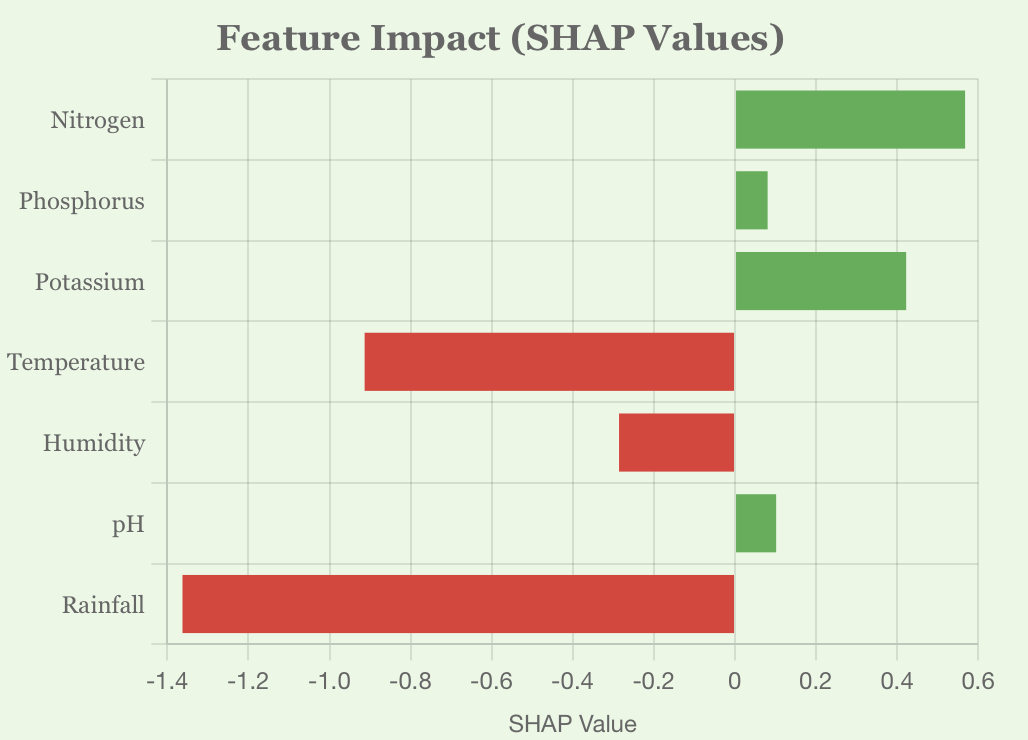


Fig. 2(b) Jute

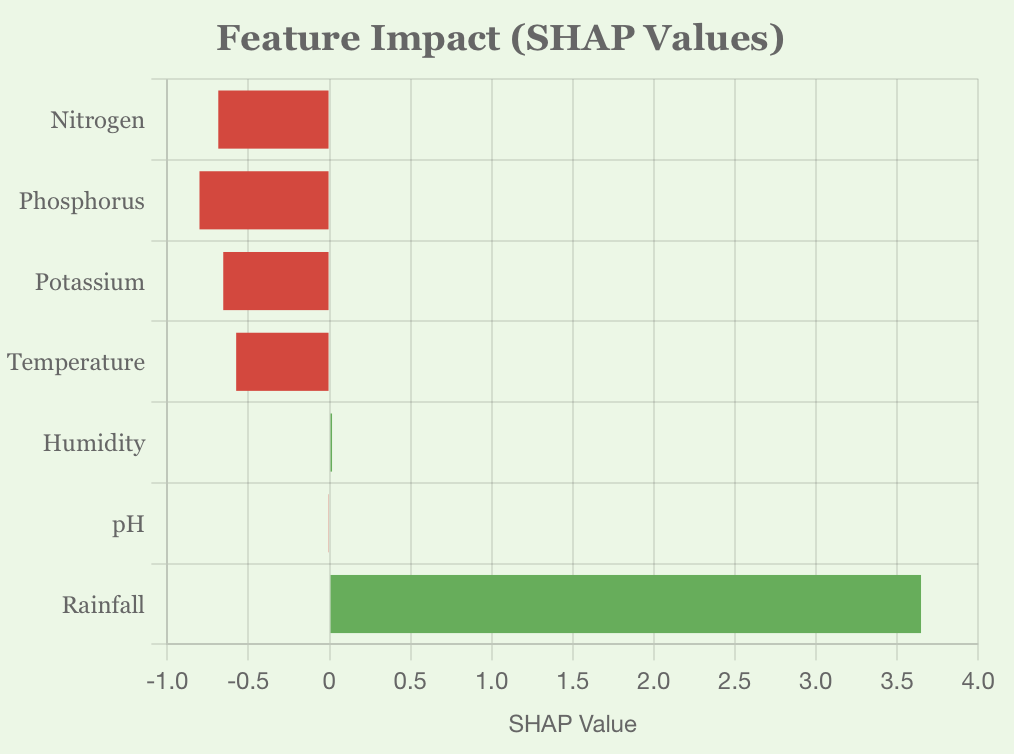


Fig. 2(c) Muskmelon

Fig. 2. SHAP Explanation Plots for top Three Recommendation

Plots in Fig. 2. reveal the exact reasoning behind a recommendation, such as high rainfall and pH supporting rice cultivation while excess potassium could hinder coffee growth. This local interpretability builds user confidence and opens avenues for agronomic discussions.

## Global Feature Attribution

Beyond per-instance explanations, we also computed the global feature importance based on SHAP values aggregated across the entire training set. Fig. 3 outlines which features have the greatest impact on crop decisions irrespective of the user input.

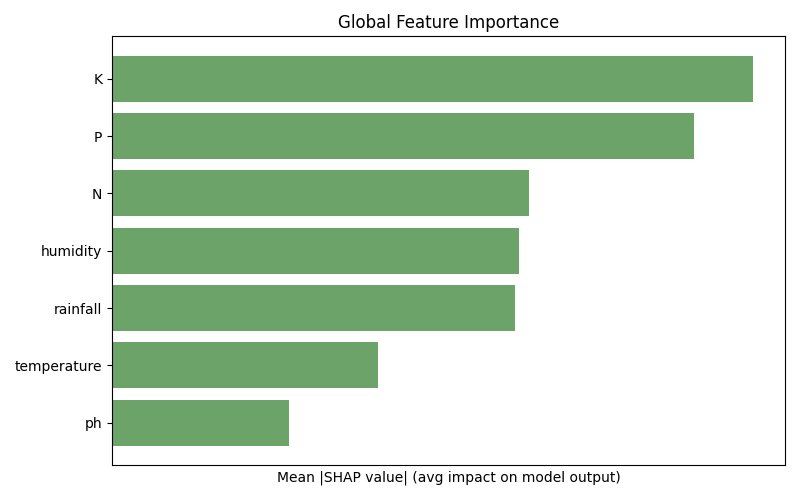


Fig. 3 Global Feature Importance

## Trust Score and Prediction Confidence

To quantify the reliability of each prediction, we introduced a Trust Score module. This score is derived from the confidence gap between the top prediction and the runner-up, normalized with a threshold margin. The output includes Trust Level (High, Moderate, Low) and Confidence Score (as a percentage) as shown in Fig. 4.

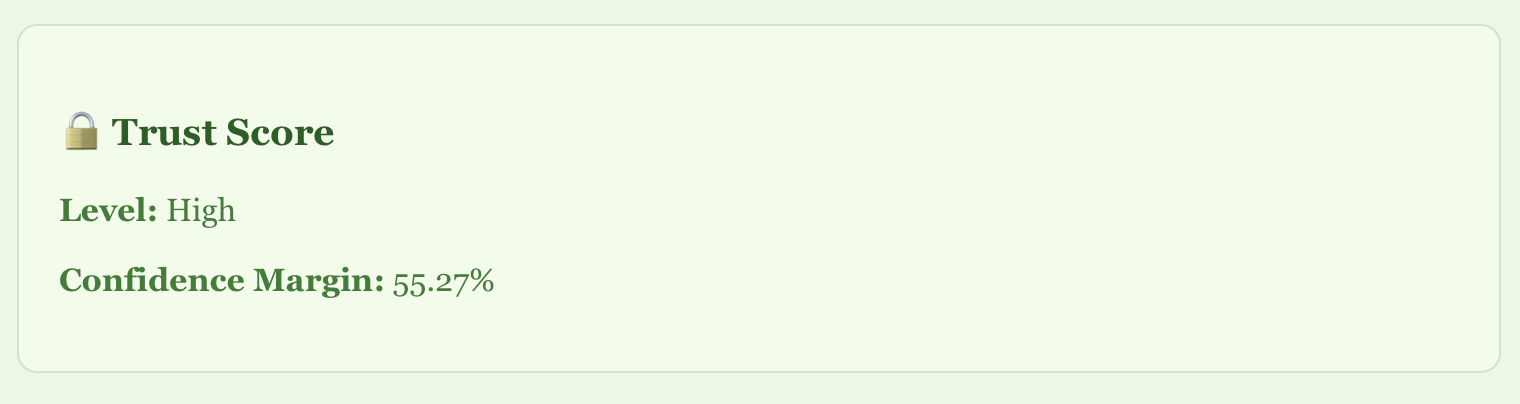


Fig. 4. Trust Score Module

## Counterfactual Explanations

Going beyond explanation, our system offers counterfactual suggestions—minimal adjustments to the controllable features (N, P, K, pH) to make another viable crop a better fit. This is particularly useful when the top crop is not feasible due to market or logistical constraints.

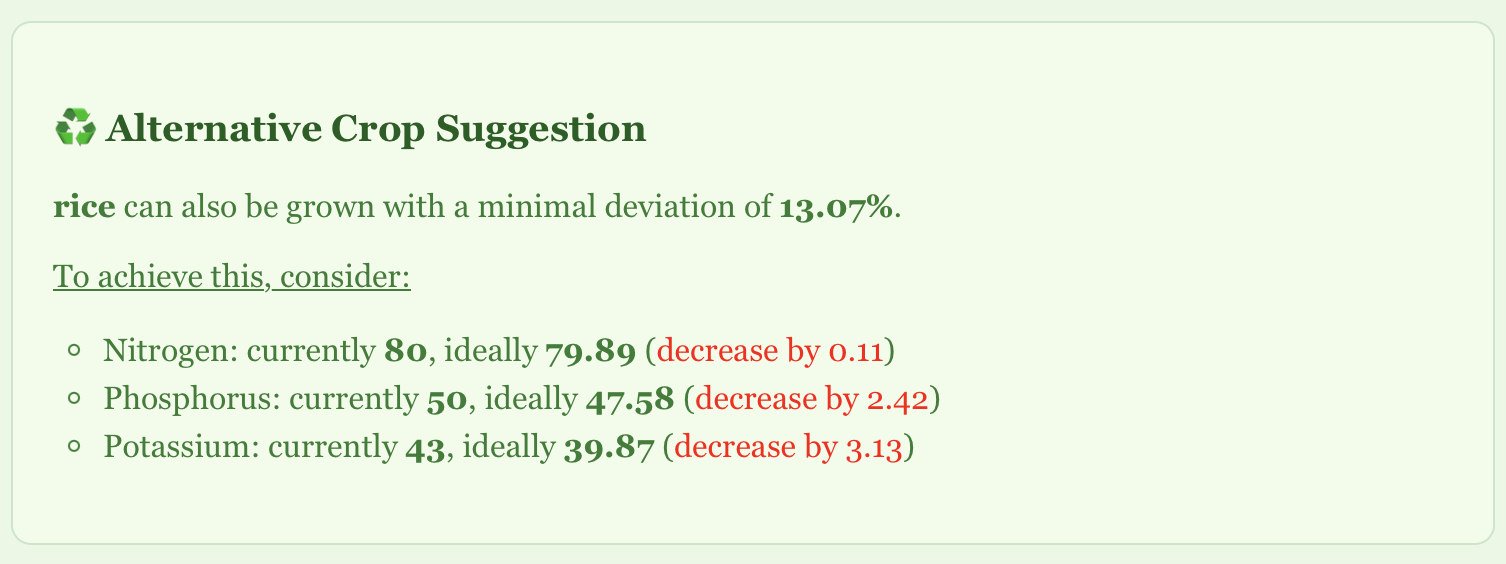


Fig. 5. Counterfactual Crop Suggestion

The system computes per-feature deviation from ideal crop conditions and identifies the next best crop that requires the least intervention. Weather-based features like rainfall or humidity are excluded from this computation to focus only on soil-amendable variables.

## System Usability and Web Deployment

To facilitate real-time access and ease of use, the entire pipeline is deployed via a web interface as shown in Fig. 6. The user enters available inputs through an intuitive form with live condition tracking. Once submitted, results are dynamically visualized and explained.

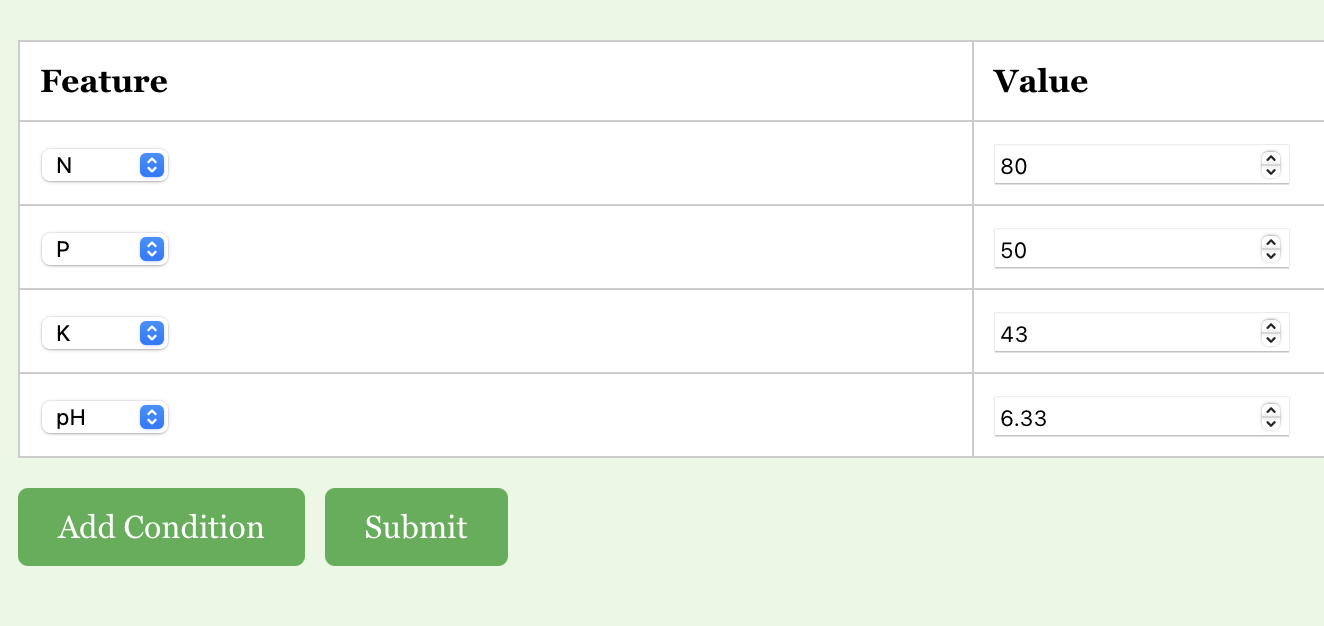


Fig. 6(a) Input



Fig. 6(b) Top three Recommended Crops

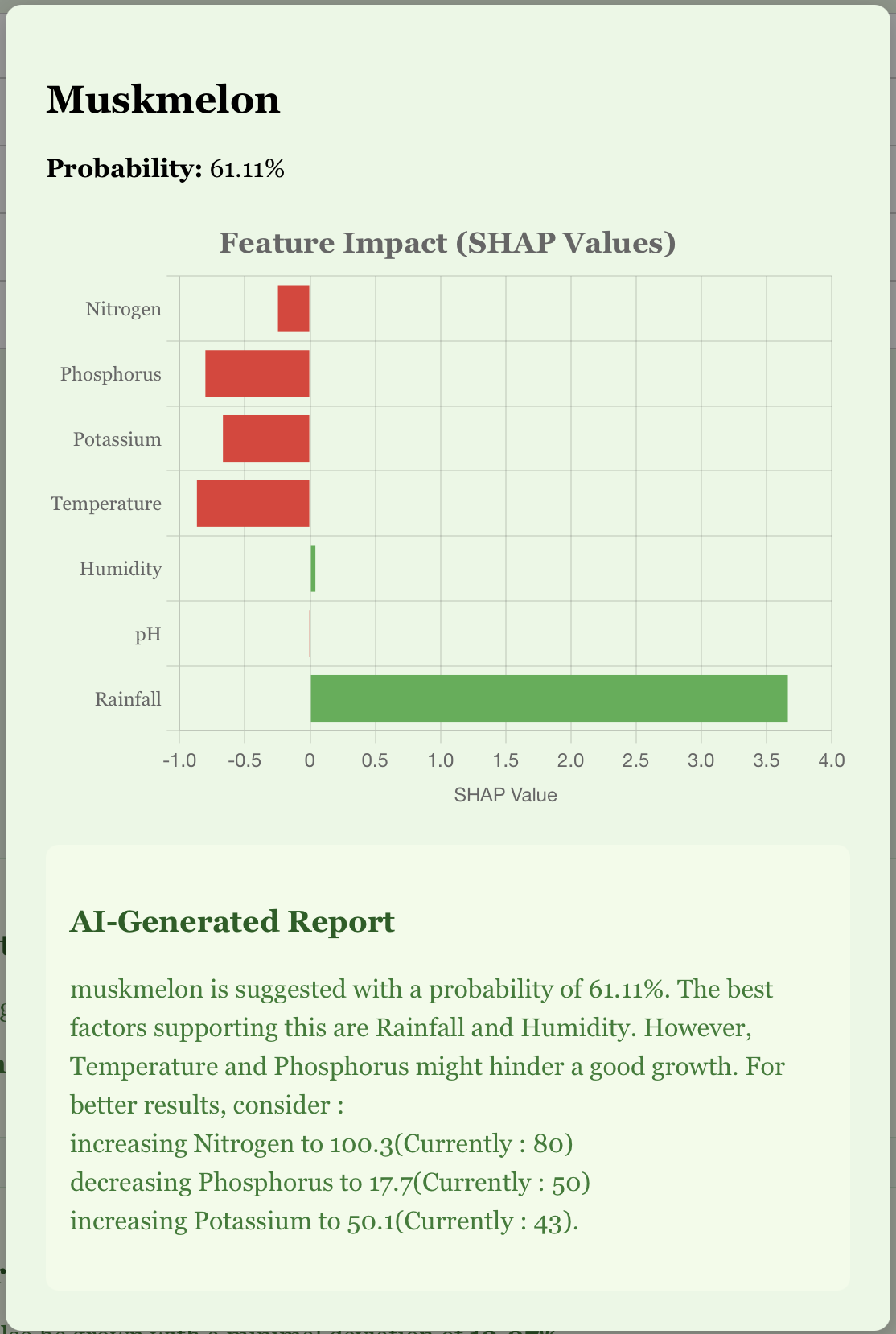


Fig. 6(c) Detailed Crop Suggestion(Modal)

# Conclusion and Future Scope

In this research, we developed and deployed a fully functional Explainable AI-powered Crop Recommendation System that bridges the gap between black-box machine learning predictions and farmer-accessible interpretability. Our pipeline, built upon an XGBoost classification model, leverages SHAP-based explainability to generate intuitive visual and textual insights, making it highly interpretable for agronomists and non-technical stakeholders alike.

The system not only predicts the top three suitable crops for a given set of environmental and soil parameters but also explains the rationale behind those predictions using local SHAP plots, global feature importance visualizations, trust scoring mechanisms, and counterfactual suggestions. By integrating real-time weather data from Open-Meteo APIs and deploying a fully responsive frontend, we provide users with a seamless and intelligent decision-support experience.

The interpretability layer was instrumental in ensuring that stakeholders understood not only what the system was recommending, but also why and how the recommendations could be improved—an essential factor in promoting trust and adoption among farming communities. While the current system offers a robust and explainable crop recommendation engine, several avenues for future enhancement exist:

1. Sensor-Based Real-Time Soil Monitoring: Future iterations of this system can be integrated with IoT-based soil sensors capable of measuring real-time Nitrogen (N), Phosphorus (P), Potassium (K), and pH values. These sensors can be deployed directly in the field, transmitting data wirelessly to the backend. This would eliminate manual data entry, reduce human error, and enhance the granularity and reliability of the input data.
2. Region-Aware Fine-Tuning: Training separate models or fine-tuning the existing one for specific agro-climatic zones can lead to more context-aware recommendations, especially for geographically diverse countries like India.
3. Fertilizer and Irrigation Planning: The model can be extended to not only suggest crops but also recommend specific fertilization plans, irrigation cycles, and crop rotation schedules tailored to user inputs.
4. Adaptive Learning from Feedback: Incorporating a feedback loop where farmers confirm or reject system suggestions can help the model retrain itself periodically, making it progressively smarter and more accurate over time.

# References

1. Jha, A., et al., “SVM-based Crop Recommendation System Using Soil and Climate Data,” *IJCA*, vol. 181, no. 14, pp. 7-12, 2019.
2. Banerjee, A., et al., “Climate-Aware Crop Prediction Using Random Forest,” *Sustainable Computing: Informatics and Systems*, vol. 29, 2021.
3. Patel, R., et al., “Crop Selection Using Hybrid ML for Small Farmers,” *IJARIIT*, vol. 5, no. 2, 2018.
4. Ali, A., et al., “Interpreting Agricultural Predictions Using LIME,” *Computers and Electronics in Agriculture*, vol. 182, 2021.
5. Hossain, M., et al., “Decision Tree Based Crop Recommendation App,” *IEEE IEMCON*, pp. 213-218, 2020.
6. Dey, A., et al., “Smart Agriculture through Weather-Driven Crop Recommender,” *IJCSMC*, vol. 9, no. 6, pp. 39-44, 2020.
7. Singh, A. and Rao, A., “Ensemble-Based Crop Predictor for Indian Farming,” *IEEE ICACCT*, 2019.
8. Kumar, N., et al., “Crop Suitability via Attention-Based Neural Models,” *Computational Intelligence*, vol. 38, no. 3, 2022.
9. Sharma, M., et al., “Fuzzy Rule-Based Crop Suitability System,” *IJRECE*, vol. 7, no. 1, 2019.
10. Yadav, R., et al., “SHAP Interpretability for Wheat Prediction,” *Elsevier Ecological Informatics*, vol. 61, 2021.
11. Brar, M., et al., “Geolocation-Aware Crop Recommender Using Deep Learning,” *IEEE Access*, vol. 10, pp. 19855–19867, 2022.
12. Das, P., et al., “Explainable Boosting Models in Agricultural Systems,” *Frontiers in AI*, vol. 6, 2023.