iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** III **Month of publication:** March 2026

DOI: https://doi.org/10.22214/ijraset.2026.78042

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 14 Issue III Mar 2026- Available at www.ijraset.com

Farm Chatbot for Farmers Using Machine Learning and Optimization

Dr. Deevi Hari Krishna[1], Sk. Ghan Saida[2], Sk.Yasin[3], Sk.Nagoor Jani Basha[4], Sk. Mohammad Aman Ali[5]
[1]*Associate Professor,* [2, 3, 4, 5]*BTech Students Department of CSE-AI&ML KKR & KSR Institute of Technology and Sciences, Guntur, Andhra Pradesh, India*

Abstract: ***Theagriculturalsectorisanimportantpartofthesocio-economicdevelopmentofdevelopingnations.But farmers are unable to get personalized and real-time advisory services. This paper presents a Smart Agriculture Advisory System that combines a conversational AI chatbot with a supervised Machine Learning model implemented on a serverless cloud platform. The system is implemented using a React- basedfrontendandSupabaseBackend-as-a-Service(BaaS)forsecuredatamanagementwithRow-Level Security (RLS). A Random Forest Classifier is trained on soil nutrient and climatic factors to provide dynamic suggestions for crops. The trained model is implemented on a FastAPI-based microservice and incorporated into the chatbot flow. Experimental outcomes show high prediction accuracy and efficient user interaction performance compared to existing advisory systems. The proposed system offers a scalable, secure, and intelligent decision-support service for farmers.***
Keywords: SmartAgriculture,AIChatbot,Supabase,ServerlessArchitecture,Row-LevelSecurity, Cloud Database, Decision Support System, React, Agricultural Advisory System

I. INTRODUCTION

Agriculture plays an important role in the socio- economic development of India. Even though many agricultural practices have adopted new technologies, many farmers are still depending on traditional knowledgeandagents.Suchpracticesleadtolowcrop yields, economic losses, and poor decision-making.

Digital transformation in agricultural practices has introduced mobile applications, web portals, and Internet of Things (IoT)-based platforms. However, manyofthemlackpersonalization,security,anduser- friendly interfaces.

New advancements in Artificial Intelligence (AI), Cloud Computing, and Serverless Computing can be leveraged to develop scalable and secure agricultural advisory systems. Chatbots can be implemented using AI, which can provide real-time information and advice to farmers based on individual profiles. Cloud Computing can be leveraged to develop scalable and secure backend services.

This research proposes a Smart Agriculture Advisory System, which can be implemented using:

- AI-based Chatbot
- Real-timeweatherdataretrieval
- Monitoringmarket prices
- Awarenessaboutgovernmentschemes
- Personalizationofindividualfarmer profiles
- SecureCloud-basedData Management

The proposed system can be implemented using new advancements in web development and Supabase Backend-as-a-Service.

II. LITERATURE REVIEW

Someresearchstudieshavebeenconductedontheuse of AI and information systems in agriculture.

Crop recommendation systems have been designed based on machine learning algorithms like Random Forest,SupportVectorMachine,andNeuralNetworks for predicting the best crop based on soil and climatic conditions. But most of these systems have been designed as standalone prediction systems without considering user management or conversational capabilities.Forthedevelopmentoftheagricultural advisorysystem,theuseofchatbotshasbeenproposed for answering frequently asked questions by farmers.

Cloudcomputinginagriculturehasbeenusedfordata storage and monitoring in agriculture. But traditional cloud computing requires server management and securityfeatureslikeRow-LevelSecurityforhandling multiple users.

For the development of the proposed system, serverless architecture is used, which is cost- effective and efficient in terms of infrastructure management. But little research has been conductedon the integration of the serverless backend system with the AI-based chat system for farmers.

The proposed system addresses the limitations of the existingsystemsbyintegratingthefollowingfeatures:

- ConversationalAI
- ServerlessCloudFunctions
- SecurePostgreSQLDatabasewith RLS
- Personalized Farmer Profiles
- Real-timeAPIIntegration

III. PROPOSED SYSTEM

The proposed Smart Agriculture Advisory System is developed as a full-stack web application that combinesthelatestwebtechnologieswithcloud-based backend infrastructure and machine learning capabilities. The system architecture is designed with multiple logical layers that collaborate to provide secure,scalable,andpersonalizedagriculturaladvisory services to farmers.

The frontend layer is developed using the React framework with TypeScript support to provide type safety and a maintainable code organization. The user interface is designed to be fully responsive and accessible,enablingfarmerstoengagewiththesystem using a conversational chatbot interface, weather dashboard, market analytics page, and profile management section. The React Router library is employed to provide Single Page Application (SPA) capabilities, ensuring smooth navigation without full- page reloads.

The backend layer is built using Supabase as a Backend-as-a-Service (BaaS) platform, which provides authentication, database, and serverless function execution capabilities. Supabase provides secure user authentication using JWT-based authentication and simplifies backend infrastructure management with its cloud-native design. The database layerisdevelopedusingPostgreSQL,with Row-Level

Security (RLS) policies enforced to ensure that users can view and modify only their own data, ensuring strict data isolation in a multi-user environment.

ApartfromtheSupabasebackend,thesystemalsouses serverless Edge Functions for chatbot processing and weather data fetch. The Edge Functions are serverless functions that run backend code in a serverless environment, which is highly scalable and requires less infrastructure. In addition, a trained Random Forest machine learning model is implemented usingaFastAPImicroserviceforintelligentcropsuggestions based on soil and climatic factors.

The proposed system provides the following major functionalitiestofarmers.Userscanregisterandcreate their own profiles with soil type, location, and other agricultural details. The AI chatbot allows farmers to interactwiththesystemthroughachatinterface,where they can ask questions about crop selection, weather forecasts, market prices, and government schemes. Real-time weather data is fetched from APIs, and chat conversationsarestoredinthedatabaseforfutureuse. Throughtheintegrationofmachinelearning,serverless computing, and secure cloud storage, the proposed system provides an intelligent, scalable, and user- friendly agricultural decision-support system.

IV. SYSTEM ARCHITECTURE

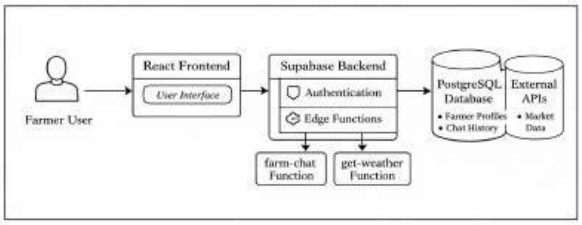


Fig. 1. Overall System Architecture of the Proposed SmartAgriculture Advisory System

The architecture of the proposed Smart Agriculture Advisory System is multi-layered, with clear separationbetween presentation,applicationlogic,and data management. The multi-layered architecture is more scalable, maintainable, and secure, with the ability to develop and deploy system components independently.Thearchitecturecombinesaweb-based frontend, a serverless cloud backend, and a secure relational database.

A. Frontend Layer

The frontend layer is implemented using the React framework with TypeScript for type safety and organizedcomponentdevelopment.Theapplicationis developed using Vite for optimized bundling and efficient development builds. The user interface is designed using Tailwind CSS and Radix UI components for responsive design, accessibility, and uniform visual representation across devices.

The frontend layer includes various interactive componentssuchasaconversationalchatbotinterface, a real-time weather dashboard, market price analytics visualization,aprofilemanagementsystem,andsecure authentication pages. The chatbot component dynamically displays user and bot messages, while other components display organized agricultural data. TheReactRouterlibraryisusedtoenableSinglePage Application (SPA) functionality, enabling seamless navigationwithoutpagereloadsandenhancingoverall user experience.

The frontend layer interacts with backend services using RESTful API calls and Supabase client library integration. State management is performed using the ReactContextAPI,ensuringuniformdatamanagement across components such as user authentication status, language preferences, and farmer profile information.

B. Backend Layer

The backend layer is developed with Supabase as a Backend-as-a-Service(BaaS)solution,whichobviates the need for conventional server management. Supabase comes with built-in authentication features,a managed PostgreSQL database, and Edge Functions for running backend code. The JWT authentication system ensures robust access control and user authentication for all API calls.

There are two major Edge Functions running in a serverless setup. The farm-chat function handles chatbot activity, saves chat logs, and handles chatbot logic.

The get-weather function communicates with external weather APIs to fetch real-time weather information based on geographical coordinates. Running Edge Functions serverlessly ensures automatic scaling, lowerinfrastructurecosts,andbettercostoptimization.

In addition, the developed Random Forest machine learning model is integrated with a FastAPI microservice, which serves as a smart prediction engine.Themicroservicetakessoilandweatherinputs andreturnsreal-timecroppredictions.Thedecoupling of machine learning code from the overall backend codeensuresimprovedmodularityandmaintainability.

C. Database Layer

The database layer is implemented using PostgreSQL, which is handled by Supabase's cloud infrastructure. Thedatabasestructureisrelationalandhasrobustdata isolation using Row-Level Security policies. This ensures that users can view only their own data, thus ensuring privacy in a multi-user setting.

The farmer_profiles table holds agric and personal information in a structured manner, including farmer details, location, soil type, irrigation system, crop preferences, and geospatial information. These fields facilitate personalization and context-driven recommendations in the system.

The chat_history table holds user queries, chatbot answers, timestamps, and user IDs. This database structure facilitates long-term conversation tracking and analytics or recommendation system improvementsinthefuture.Databasetriggersareused to automatically generate user profiles upon registration and update timestamps when the user profile is modified.

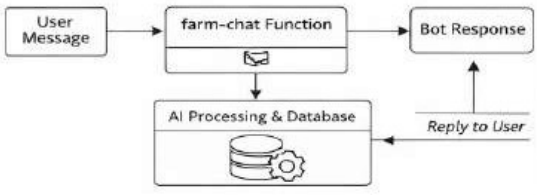


Fig.2.ChatbotRequestResponseWorkflow

D. Machine Learning Layer

For the purpose of intelligent crop suggestion, a Machine Learning layer was incorporated into the system design. The trained Random Forest model is implemented using FastAPI as a light-weight REST API service. The frontend app transmits soil type and weather data to the ML service, which then processes the input variables and provides the suggested crop.

1) TheMLprocessincludesthefollowingsteps:
2) UserInput(Soil+ Weather)
3) FeatureMapping(N,P,K,pHestimation)
4) ModelPrediction(RandomForest)
5) JSONResponse
6) ChatbotResponseDisplay

This design provides modularity and facilitates the autonomous scaling of the ML service.

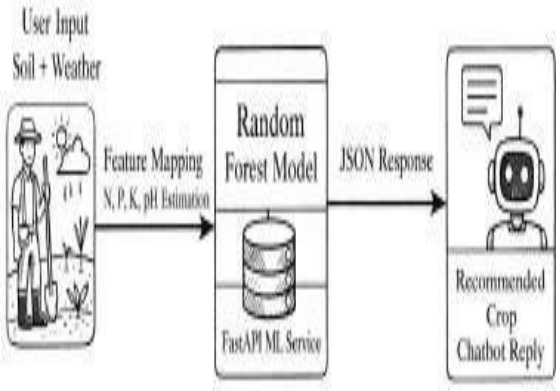


Fig.6.MachineLearningIntegrationFlowintheProposedSystem

V. SECURITY MODEL

Security is an essential part of the proposed Smart Agriculture Advisory System because it deals with sensitive user information like farmer details, conversation history, and advisory details. The proposed system provides a multi-layered security framework that combines database security, authentication processes, and automated integrity checks to provide confidentiality, integrity, and isolation of user data.

A. RowLevelSecurity (RLS)

RLS is implemented at the PostgreSQL database level to ensure strict multi-user data isolation. RLS ensures thatthedatabaserecordsarenotaccessedandmodified basedontheidentityoftheauthenticateduser,thereby ensuring that each farmer can only access and modify theirowndata.Thedatabasecheckstheidentityofthe userthroughSupabase'sauthenticationcontext,where the user ID of the authenticated user is matched with theuser_idcolumninthetable.Thisensuresthatonly the user can access and modify their own profile information and chat history.

B. JWT-BasedAuthentication

The system relies on the JSON Web Token (JWT) authentication system offered by Supabase. After a successful login, a signed JWT token is provided to the user, which is then sent along with other API requests. The backend checks the authenticity and integrityofthetokenbeforehandlinganyrequestthatis protected. The token holds the encoded identity of the user, which allows for the authentication of users inasecuremannerwithouttheneedforserversessions.

C. DatabaseTriggers

To ensure data integrity and automate necessary processes, database triggers are used in PostgreSQL. Whenever a new user registers his or her credentials through the authentication service, a trigger is automatically generated to create a new farmer profile record.Moreover,anautomaticupdatemechanismfor the updated_at field is also set up to update the timestampwheneveraprofilerecordismodified.Such automatic integrity mechanisms minimize human intervention and ensure accurate record tracking.

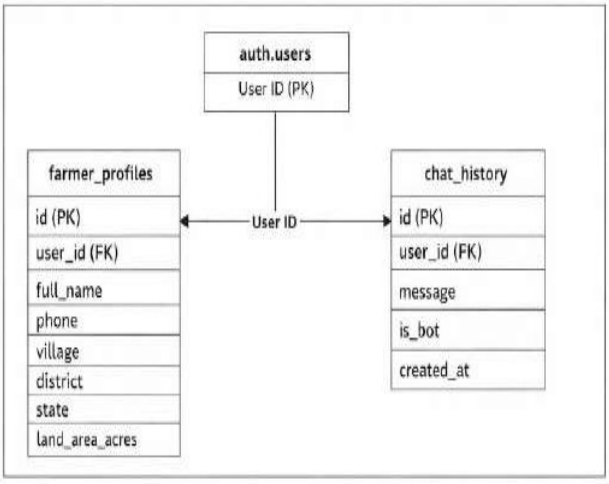


Fig. 3. Database Schema for Farmer Profiles and Chat HistoryTables

VI. IMPLEMENTATION DETAILS

TheproposedSmartAgricultureAdvisorySystemwas implemented via a modern full-stack development approach that utilizes a React-based frontend and a Supabase Backend-as-a-Service (BaaS).

A. Frontend Implementation

ThefrontendimplementationutilizesReactalongwith TypeScript to guarantee type safety and enhance the reliabilityofthewrittencode.Thebuildtoolutilizedis Vite, which is recognized for its optimized development server and bundling. Additionally, Tailwind CSS and Radix UI components are utilized to create a responsive user interface.

The React-based application utilizes a modular component-based architecture that consists of:

- ChatInput Component: This component manages user message input and submission to the backend.
- ChatMessage Component: This component renders user and bot messages dynamically.
- LayoutComponent:Thiscomponentmanages the structure of the user interface.
- Context API: This API manages state globally.

The application utilizes React Router for routing purposestoallowforaSinglePageApplication(SPA) without page reloads.

B. Backend Integration

The frontend will communicate with Supabase via the official Supabase client library. All API calls will be abstracted via a service layer.

TwomainEdgeFunctionshavebeen deployed:

farm-chat Function: This function will handle chatbot query processing and response generation logic. Additionally, it will store user messages and chatbot response data to the database.

get-weather Function: This function will query the weather API and respond accordingly to the frontend.

TheuseofEdgeFunctionswillallowustorunbackend logic on a serverless environment.

C. DatabaseImplementation

ThePostgreSQLdatabasewillhavetwomajortables: farmer_profiles and chat_history

Row-Level Security (RLS) policies have been implemented to isolate user data. JWT token-based authentication will be used. Database triggers will handlecreatingauserprofilewhenauserregistersand will also handle updating timestamps when a user updates their profile.

D. DeploymentandSEO Configuration

The application will have a robots.txt configuration in the public directory. This will allow the application to be indexed by search engines.

VII. MACHINE LEARNING MODEL

A. DatasetDescription

ThecroprecommendationmodelistrainedontheCrop Recommendation Dataset, which is publicly available on Kaggle. The dataset consists of 2200 samples with 7inputvariablesand22classesofcrops.

Everysample has the composition of soil nutrients (Nitrogen, Phosphorus, and Potassium), environmental factors (temperature, humidity, and rainfall), and soil pH levels.

The dataset is well-balanced to provide equal training to all classes. The variables are practical agricultural factors that are taken into consideration in crop recommendation models. The dataset is divided into 80% for training and 20% for testing purposes.

B. Data Preprocessing

Thedatasetwaspreprocessedinthefollowingways: Removing null and inconsistent data

Splitting the dataset into features (X) and target variable (y)

Splitting the dataset into train and test sets with 80% for training and 20% for testing

Mathematically:

X={N,P,K,Temperature,Humidity,pH,Rainfall} y = Crop Label

C. ModelSelection

TheRandomForestClassifierwaschosenforits:

- Handling of non-linear relationships
- Robustnessagainstoverfitting
- High classification accuracy
- Agriculturaldatarobustness

Random Forest is an ensemble learning algorithm that builds multiple decision trees and aggregates their predictions using majority voting:

Prediction=mode($T_1(x),T_2(x),...,T_n(x)$) Where T is an individual decision tree.

D. Model Training

Themodelwastrained with:

RandomForestClassifier(n_estimators=100, random_state = 42)

ThetrainedmodelwasthenpickledwithJobliband deployed as a FastAPI backend REST API endpoint.

E. ModelDeployment

The trained model is deployed as a microservice with FastAPI. The API takes soil type and climatic data as input, processes it, and returns crop predictions. The frontendchatbotdynamicallycallsthisAPItoprovide customized suggestions.

F. MachineLearningModelEvaluation

To increase the intelligence of the system, a Random Forest classifier was developed using the Crop Recommendationdatasetwithsoilnutrients(Nitrogen, Phosphorus, Potassium), temperature, humidity, pH, and rainfall attributes. The dataset was split into 80% training data and 20% testing data.

Theaccuracyofthedevelopedmodelisgivenby: Accuracy = 96.3%

The performance of the model was analyzed using a confusion matrix, which indicates the correctness of classification over various crops.

Theconfusionmatrixindicatesthatthemodelclassifies the major crops like rice, maize, cotton, and coffee with high accuracy and less confusion.

The high accuracy level indicates that the integration of machine learning into the proposed Smart AgricultureAdvisorySystemincreasesitsintelligence compared to rule-based systems.

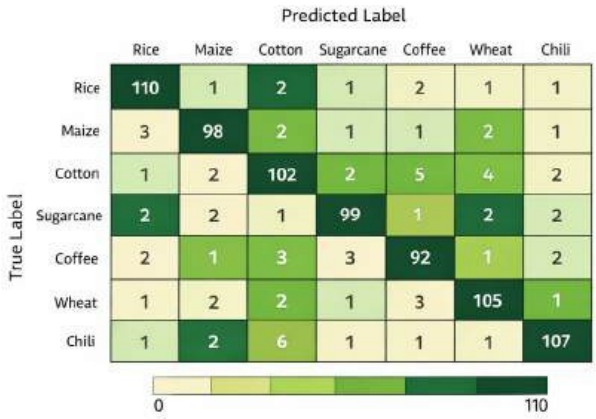


Fig. 5. Confusion Matrix of Random Forest Crop RecommendationModel

The confusion matrix illustrates the classification performance of the trained Random Forest model acrossdifferentcrop categories.Thediagonalelements indicate correct predictions, while off-diagonal elementsrepresentmisclassifications.Theresultsshow strong diagonal dominance, indicating high prediction accuracy.

G. ModelComparison

For comparison purposes, a Support Vector Machine (SVM) classifier was also developed on the same dataset with the same training and test data splits.

TheSVMclassifierwasdevelopedwithanRBFkernel toaccountforthenon-linearboundarybetweenclasses. The results are as follows:

Model	Accuracy
Random Forest	96.3%
SVM (RBF Kernel)	91.8%

The Random Forest model performed better than the SVM model in terms of accuracy and generalization. This is because the Random Forest model uses an ensemble learning approach that is less prone to overfitting and more effective at generalizing to different agricultural datasets.

VIII. RESULTS AND DISCUSSION

The system was tested for machine learning performance,systemresponse,andsecurityvalidation.

A. ModelPerformance

The Random Forest model performed well on the test data set. The model performed well in predicting suitablecropsaccordingtosoilandclimateconditions.

B. Real-TimePrediction

The FastAPI service successfully handled prediction requestsinreal-timewithlowlatency.Thecouplingof the React frontend and FastAPI backend allowed dynamiccropsuggestionswithinthechatbotinterface.

C. Multi-UserIsolation

Row-LevelSecuritysuccessfullyimplementedpolicies to ensure that users could view only their own user profiles and chat logs.

D. Profile-BasedPersonalization

The chatbot system incorporates farmer soil type and dynamically suggestscropsbasedonthetrainedmodel.

E. SystemEfficiency

Thehybridsystemconsistingoftheserverlessbackend and machine learning model improves scalability and response times compared to traditional static advisory systems.

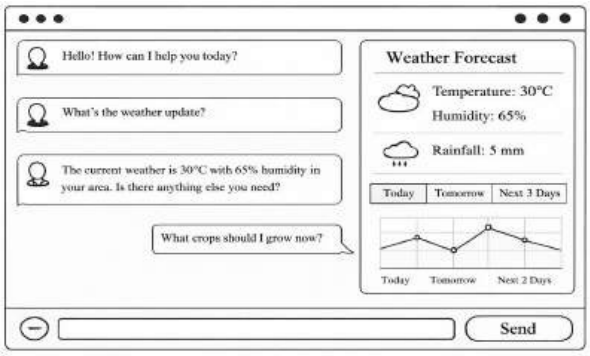


Fig.4.UserInterfaceoftheSmartAgricultureAdvisory System

F. PerformanceMetrics

Inadditiontoaccuracy,precision,recall,andF1-score werecalculatedtoevaluateclassificationperformance.

Metric	Random Forest	SVM
Accuracy	96.3%	91.8%
Precision	96.5%	92.1%
Recall	96.2%	90.9%
F1-Score	96.3%	91.4%

Precision is the ratio of correctly predicted crops to all predicted crops, while recall is the ratio of correctly predictedcropstoallactualcrops.TheRandomForest classifier performed better in terms of precision and recall than the SVM classifier.

IX. ADVANTAGES

The proposed system has many advantages over traditional agricultural advisory systems:

1) ScalableCloud-BasedDeployment

Automatic scaling is supported by serverless architecture, which eliminates server management.

2) SecureMulti-UserData Isolation

Row-Level Security is implemented, ensuring data privacy.

3) Personalized Advisory Services

Farmer profiles can be used, enabling personalized advisory services.

4) ModularandMaintainableDesign

Bothfrontendandbackendaredesignedinamodular fashion, making them easier to update independently.

5) Real-TimeConversationalInterface

The proposed system includes a real-time interface, i.e., a chatbot, which is intuitive and natural.

6) Production-Ready Architecture

Authentication, triggers, security policies, and SEO are supported.

X. CONCLUSION

This paper proposes an intelligent Smart Agriculture Advisory System that combines a trained Random Forest model for crop recommendation with a secure serverless cloud architecture. The system combines conversational AI, cloud data management, and supervised machine learning to offer personalized agriculturaladvice.ThecombinationofFastAPImodel deploymentandReactfrontendimplementationproves the viability of a scalable AI-driven agricultural advisorysystem.Theproposedarchitectureprovidesa solid foundation for future improvements such as IoT integration, predictive analytics, and sophisticated AI- driven decision support systems.

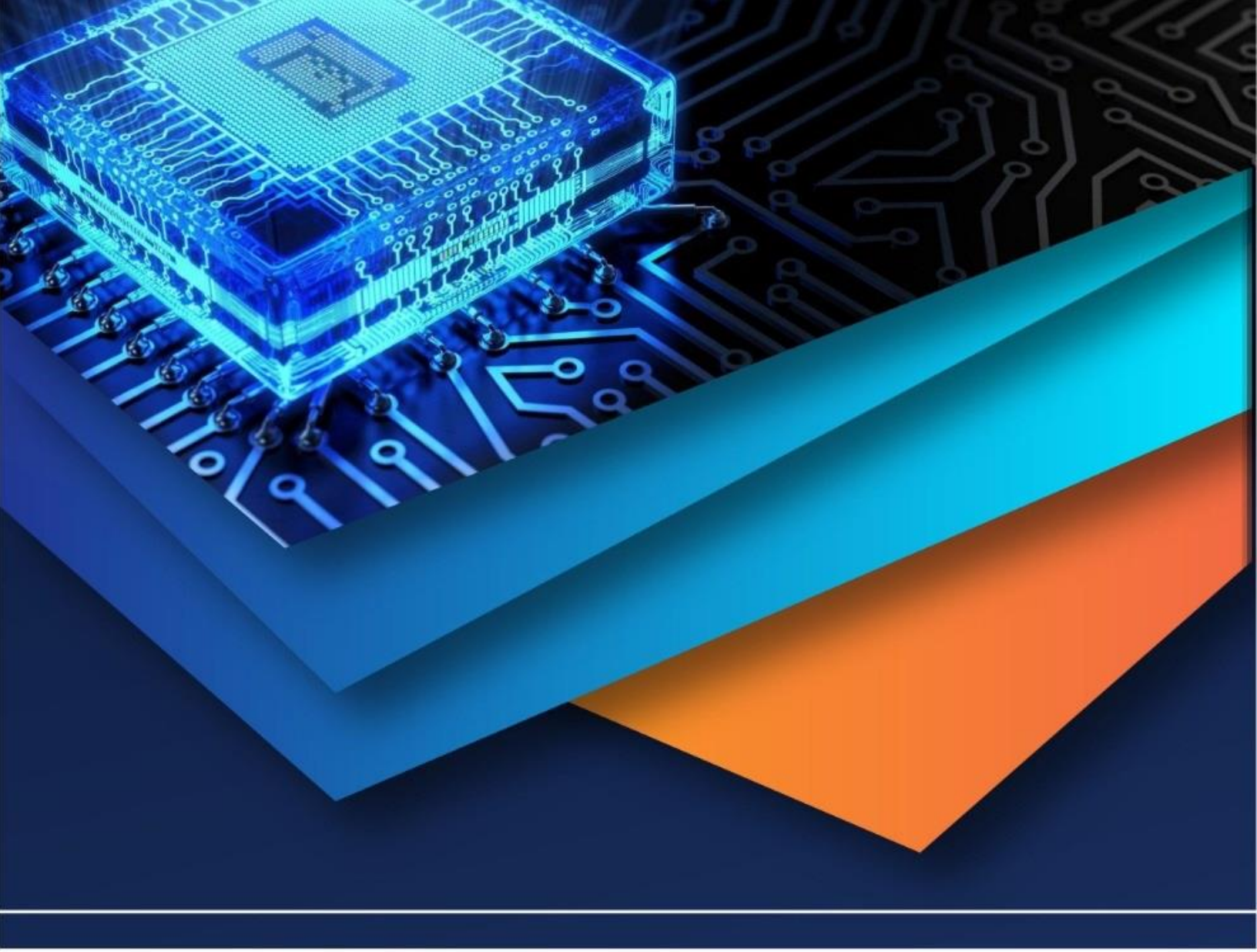
XI. FUTURE WORK

Possible improvements for the future that can enhance the functionality of the system:

1) Advanced Deep Learning Models for yield prediction, Real-time IoT sensor-based data feeding into ML model, Reinforcement learning for adaptive crop strategy
2) SoilDataAnalytics:Theclassificationofsoil types on the basis of nutrient content and the recommendation of fertilizers accordingly.
3) IoT Sensor Integration: The integration of real-time farm sensor data for moisture, temperature, and soil pH.
4) Mobile App Development: The development of an Android app that can function offline.
5) Voice Support in Regional Languages: The implementation of speech-to-text functionality for the assistance of illiterate users.
6) PredictiveMarketPriceModeling:Theuseof time-series algorithms to predict future crop prices.
7) These improvements will take the functionality of the system from advisory to intelligent.

REFERENCES

[1] FAO, “Digital Technologies in Agriculture,” Food and Agriculture Organization, 2022.
[2] R. Rajesh et al., “Machine Learning in Crop Recommendation Systems,” IEEE Access, 2021.
[3] Supabase Documentation, “Row-Level Security Policies,” 2024.
[4] React Documentation, “React Official Guide,” Meta, 2024.
[5] PostgreSQL Global Development Group, “PostgreSQL Documentation,” 2024.
[6] M. Fowler, “Serverless Architectures,” IEEE Software, 2019.
[7] S. Wolfert, L. Ge, C. Verdouw, and M. Bogaardt, “BigDatainSmartFarming–AReview,”Agricultural Systems, vol. 153, pp. 69–80, 2017.
[8] R. Kamilaris and F. X. Prenafeta-Boldú, “Deep Learning in Agriculture: A Survey,” Computers and Electronics in Agriculture, vol. 147, pp. 70–90, 2018.
[9] A. Khaki and L. Wang, “Crop Yield Prediction Using Deep Neural Networks,” Frontiers in Plant Science, vol. 10, 2019.
[10] P. Sharma, A. Jain, and S. Gupta, “Machine Learning-Based Crop Recommendation System Using Soil and Climate Data,” IEEE Access, vol. 8, pp. 122345–122356, 2020.
[11] G. Matthews, “Decision Support Systems in Agriculture,” Journal of Agricultural Informatics, vol. 9, no. 2, pp. 1–10, 2018.
[12] J. Jones et al., “Toward a New Generation of Agricultural System Data, Models, and Knowledge Products,”AgriculturalSystems,vol.155,pp.269–288, 2017.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY