

SmartFarm Assist: A Mobile and Web-Based Farm Assistant System with AI Support

Patel Mohmed Amin Ismail, Mr. Rakesh Tanwar, Mr. Srikant Singh*

School of Engineering, P P Savani University, Surat , India

*Corresponding Author

Abstract: *Agriculture continues to be the main eco-nomic force of India, but farmers who work in rural regions face difficulties because they cannot obtain vital market information or agricultural expertise or contemporary farming methods. The researchers developed SmartFarm Assist as a unified agricultural support system which functions through mobile devices and web platforms using React Native (Expo) and Node.js and Express.js and MongoDB technologies. The system provides farmers with real-time market prices sourced from India's Data.gov.in API, AI-generated pest alerts and irrigation advisories powered by the Groq LLaMA-3 language model, live weather forecasts with location-based alerts, digital farm record management, government scheme discovery, and agricultural news aggregation. The administrative web dashboard built with React.js and Chart.js allows users to see interactive data visualizations which present essential performance indicators of the system. This system allows administrators to manage user accounts, analyze agricultural patterns, and use Expo push notifications to send emergency alerts across specific areas. The proposed system bridges the digital divide in Indian agriculture by combining cloud-hosted services, AI, and mobile technology into a single farmer-centric platform. SmartFarm Assist provides more extensive capabilities than existing IoT-based systems and web-only platforms which researchers have studied in the current academic literature.*

Keywords: Smart Agriculture, AI Advisory, React Native, Node.js, Market Prices, Farm Management, MongoDB, Groq LLM, Push Notifications, Web Dashboard

I. INTRODUCTION

Agriculture is the primary livelihood for over 60% of India's rural population and contributes approximately 18% to the national GDP [1]. Despite its significance, Indian farmers routinely struggle with fragmented access to real-time market data, lack of personalized agronomic guidance, delayed aware-ness of extreme weather events, and limited knowledge of government support schemes. Traditional approaches to bridg-ing these gaps-such as extension workers, radio broadcasts, and paper-based records-are slow, geographically restricted, and unable to scale to the needs of millions of smallholder farmers. The proliferation of affordable smartphones and ex-panding 4G/5G connectivity in rural India has created a unique opportunity to deploy mobile-first agricultural platforms. The Internet of Things (IoT) and Artificial Intelligence (AI) have transformed precision agriculture globally [2-4], and cloud-native architectures now make it feasible to deliver sophisticated services to farmers without requiring expensive on-site hardware. This paper introduces SmartFarm Assist, a comprehensive mobile and web-based farm assistant system designed specifically for Indian farming conditions, with initial deployment focused on Gujarat. The mobile application, built with Expo (React Native), provides a farmer-facing interface offering market prices, AI crop advisory, weather data, farm record keeping, news, and scheme discovery. The companion web application, developed in React.js, serves as an admin-istrative dashboard for agricultural officers, cooperative man-agers, and government bodies to oversee farmer activity, dis-patch regional alerts, and analyze crop distribution trends. The key innovations of SmartFarm Assist over prior art include: (1) integration of a large language model (Groq LLaMA-3) for context-aware, location-specific pest and irrigation advice; (2) real-time market price ingestion from the official Data.gov.in API with analytics and trend visualization; (3) a push noti-fication system (Expo) enabling targeted regional broadcasts; and (4) a unified platform that combines



advisory, records, market intelligence, and administration in a single ecosystem. This paper is organized as follows. Section II reviews related work. Section III presents the system design and architecture. Section IV describes the implementation and testing. Section V provides results and discussion. Section VI concludes with recommendations for future work.

II. REVIEW OF RELATED WORK

The convergence of IoT, mobile computing, and AI in agriculture has attracted significant research attention. This section reviews representative systems to identify gaps that SmartFarm Assist addresses. Significant research has been conducted in the field of smart agriculture. Patel et al. [16] proposed a Python-based approach for paddy leaf disease detection using image processing techniques. Similarly, Chauhan et al. [24] and Purani et al. [25] introduced innovative diagnostic methods that integrate computational intelligence with traditional plant pathology.

The integration of Internet of Things has further enhanced agricultural monitoring systems. Mehta et al. [18] reviewed IoT-based solutions for real-time disease detection in rice crops, while Sinha et al. [26] developed a smart agriculture system using MQTT protocol for efficient communication. Singh & Sharma [23] proposed a novel architecture for monitoring and predicting rice plant diseases, demonstrating the effectiveness of hybrid AI models.

Supporting these applications, big data analytics plays a crucial role in handling large-scale agricultural and financial datasets. Singh [19] and Shrivastava & Singh [22] discussed various aspects of big data processing, while Singh [20] highlighted critical privacy concerns associated with large datasets. Additionally, predictive analytics techniques have been applied in diverse domains, such as gaming performance prediction by Singh et al. [21], showcasing the versatility of machine learning models.

Recent studies, including Sharma et al. [17] and Navadiya & Singh [17], emphasize the importance of integrating AI, IoT, and image processing techniques for developing efficient and scalable solutions. Overall, the literature indicates a strong convergence of intelligent technologies across domains, enabling improved accuracy, automation, and real-time decision-making.

A. IoT-Based Smart Agriculture Management (SAMS)

Nagaraja et al. [2] proposed a Smart Agriculture Management System (SAMS) that uses multiple sensor nodes to measure soil moisture, temperature, and humidity and relay data to a web portal. The system successfully automates irrigation triggers based on sensor thresholds and provides a farmer dashboard. However, the system lacks a mobile application for field access, does not offer AI-powered advisory, and provides no market price integration. Its reliance on physical sensor infrastructure also limits scalability for resource-constrained farmers.

B. Raspberry Pi-Based Irrigation Automation

Rao and Sridhar [3] developed an IoT system using Raspberry Pi to automate crop irrigation by continuously monitoring soil moisture. Their work optimizes water consumption for individual crops but focuses narrowly on irrigation without addressing other critical farming needs such as price discovery, weather alerts, or scheme awareness. The absence of a smart-phone interface further restricts its utility for rural farmers who rely primarily on mobile devices.

C. iFarming Livestock System

The iFarming platform [4] integrates automated climate control, feeding systems, and continuous animal behavior monitoring for livestock farming. While effective for large commercial operations, the platform's cost, complexity, and hardware dependencies make it inaccessible for smallholder and mixed-crop farmers. It also offers no market intelligence or government scheme integration.



D. Arduino-Based IoT Farm Monitoring

Banahene et al. [5] presented an IoT farm monitoring system using Arduino microcontrollers, ESP8266 Wi-Fi modules, and a cloud database (Google Firebase). The system monitors greenhouse temperature, soil moisture, humidity, and water levels, and an LSTM model achieves 95.3% accuracy on temperature prediction. While the LSTM prediction capability is noteworthy, the system requires physical sensor deployment, provides no mobile application, lacks market data integration, and does not offer AI language model advisory or push notification capabilities.

E. Summary of Gaps

Table I summarizes the comparative analysis. Most existing systems are IoT-centric, require sensor hardware, and provide narrow functionality (irrigation or temperature monitoring). None of the reviewed systems combine AI LLM advisory, real-time national market prices, government scheme discovery, digital farm records, and push notifications in a unified mobile-and-web platform. SmartFarm Assist is designed to address this complete feature gap.

III. SYSTEM DESIGN AND METHODOLOGY

SmartFarm Assist follows a three-tier client-server architecture comprising (1) client applications (mobile and web), (2) a RESTful API server, and (3) a cloud database with integrated third-party services. The design prioritizes accessibility for low-bandwidth environments, secure role-based access control, and modular extensibility. System Architecture

Figure 1 illustrates the overall architecture. The mobile application communicates with the Express.js API server over HTTPS using JSON Web Tokens (JWT) for authentication. The admin web dashboard interacts with the same API but is protected by a separate adminAuth middleware layer. The Express.js server integrates MongoDB for persistent storage and connects to several external APIs: OpenWeatherMap and WeatherAPI for meteorological data, Data.gov.in for agricultural market prices, Groq SDK for AI language model queries, and Expo Push Notification Service for mobile alerts.

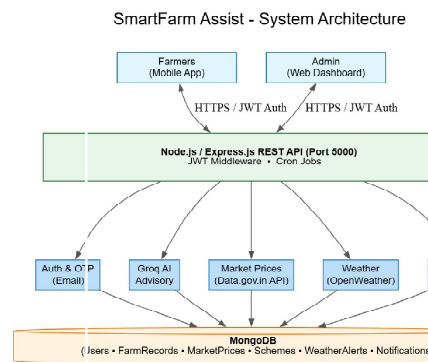


Fig. 1. SmartFarm Assist System Architecture Diagram

A. System Modules

The system is organized into ten functional modules. Table II provides a summary of each module, its underlying technology stack, and its primary functionality.

B. Database Design

MongoDB was selected as the database for its flexible document model, which accommodates the heterogeneous data structures common in agricultural datasets (e.g., variable crop lists, nested fertilizer records, multi-value sensor readings). The primary collections are: Users (farmer profiles with location, farming type, and major crops), FarmRecords (per-crop records including sowing date, harvest date, fertilizer, irrigation, and expected yield), MarketPrices (daily mandi prices per commodity and district), Schemes (government scheme metadata and application



links), WeatherAlerts (auto-generated regional weather warnings), NotificationTokens (Expo push to-kens per user), and Notifications (dispatch history for broadcast messages).

The FarmRecord schema captures fields for crop name (constrained to an enumerated set), season (Kharif/Rabi/Zaid), land area in acres, fertilizer details (N-P-K ratio, quantity, application method), and irrigation parameters (method, frequency, and volume). This structured schema enables the crop analytics module to perform MongoDB aggregation pipelines computing top cultivated crops by area, farmer count per crop, and district-wise crop distribution.

TABLE I: COMPARATIVE ANALYSIS OF RELATED WORK

| Feature | SAMS [2] | RasPi Irrigation [3] | iFarming [4] | IoT Farm [5] | SmartFarm Assist (Ours) |
|-------------------------|----------|----------------------|--------------|--------------|-------------------------|
| Mobile Application | × | × | ✓ | × | ✓ |
| Web Dashboard | ✓ | ✓ | ✓ | ✓ | ✓ |
| AI-based Advisory | × | × | × | × | (Groq LLM) |
| Real-time Market Prices | × | × | × | × | |
| Government Schemes | × | × | × | × | |
| Weather Alerts | × | | × | × | |
| Farm Record Keeping | Partial | × | × | Partial | |
| Push Notifications | × | × | × | × | (Expo) |
| SMS Alert | × | × | × | | × |
| IoT Sensor Integration | | | | | × |
| Crop Analytics | × | × | × | × | |
| Admin Panel | × | × | | × | |

TABLE II: SMARTFARM ASSIST - SYSTEM MODULES OVERVIEW

| Module | Technology / API | Functionality |
|--------------------|-----------------------------|--|
| Authentication | JWT, Bcrypt, Node-mailer | OTP email verification, secure login, password reset |
| AI Advisory | Groq SDK (LLaMA 3) | AI pest alerts and irrigation tips based on location & crops |
| Market Prices | Data.gov.in API, MongoDB | Real-time mandi prices with analytics and trend charts |
| Weather Module | OpenWeather API, WeatherAPI | 7-day forecasts, weather alerts, regional notifications |
| Farm Records | MongoDB, Mongoose | Crop tracking, fertilizer, irrigation logs per farmer |
| Admin Dashboard | React JS, Chart.js | Farmer statistics, crop analytics, region alert dispatch |
| News & Schemes | RSS Parser, MongoDB | Agri-news aggregation, government scheme browser |
| Push Notifications | Expo Server SDK | Targeted notifications to farmers by region or crop |
| Crop Analytics | MongoDB Aggregation | Top cultivated crops, district-wise crop distribution |
| Divisions Module | React, REST API | Geographic division management for ad-min |

C. AI Advisory Module

The AI Advisory module uses the Groq SDK to interface with Meta's LLaMA-3 large language model. When a farmer requests pest alerts or irrigation advice, the system retrieves the current weather conditions for the farmer's GPS coordinates from the OpenWeather API. This weather context-temperature, humidity, wind speed, and precipitation forecast-is combined with the farmer's crop list and growth stage into a structured prompt sent to the Groq API. The



model returns a JSON-formatted advisory containing identified risks, recommended actions, and application timing. The system parses this JSON and presents the advice in a readable card format within the mobile application. This approach enables the system to deliver personalized, context-aware advice without requiring a dedicated agronomic ML model hosted on-premise.

D. Market Price Module

Real-time mandi (agricultural market) prices are retrieved from the official Data.gov.in commodity prices API using the API key registered under the project. The backend caches prices in MongoDB with daily refresh cycles managed by a node-cron job. The admin panel includes a Market Analytics page built with Chart.js showing price trends over time, com-mo-dity comparisons by district, and price volatility indicators. Farmers can filter prices by commodity, state, district, and market name through the mobile interface.

E. Weather and Alert Module

The Weather module provides 7-day forecasts, current con-conditions, and hourly breakdowns using the OpenWeather and WeatherAPI endpoints. An automated weather alert job (node-schedule) runs at configurable intervals, checks for extreme weather conditions (high winds, heavy rainfall, frost, or heat-wave) in each registered district, and creates WeatherAlert documents when thresholds are exceeded. The Regional Alert system in the admin dashboard allows administrators to com-pose custom alerts and broadcast them to all farmers in a selected district via Expo push notifications.

F. Mobile Application Design

The mobile application is built with Expo (React Native), targeting Android and iOS. The authentication flow uses email-based OTP verification via Nodemailer (Gmail SMTP). After login, a bottom tab navigator provides access to the Dashboard, Market Prices, AI Advisory, Farm Records, News, and Schemes screens. The Weather screen features an ani-mated display of current conditions, a 7-day forecast strip, and a precipitation probability chart. The Farm Records screen allows farmers to log new crop cycles and view historical records with filterable sorting. All screens support both light and dark themes.

G. Admin Web Dashboard Design

The Admin Dashboard is a single-page React application with a collapsible sidebar and dark/light theme toggle. The Dashboard page presents stat cards (total farmers, recent sign-ups, total schemes, notifications sent) and three Chart.js visual-izations: farmer growth line chart (by month), scheme category distribution (doughnut), and most cultivated crops (bar). The Farmers page renders a paginated, searchable table of all registered users with farm detail expansion. The Divisions page maps farmer distribution across geographic divisions. The Farm Records page enables admins to review crop records system-wide. The Market Analytics and Market Prices pages provide detailed commodity intelligence for administrative use.

IV. SYSTEM IMPLEMENTATION AND TESTING

A. Development Environment

The backend was developed using Node.js v18 with the Express.js framework and runs on port 5000. Mongoose v8 is used as the ODM layer for MongoDB. The admin frontend was bootstrapped with Create React App (React 18) and uses Bootstrap 5 alongside custom CSS variables for theming. Nodemon is used in development mode for hot-reloading. The mobile application was developed in Expo SDK 51 with Expo Router for file-based navigation.



B. Authentication and Security

User authentication uses bcryptjs for password hashing with a salt factor of 12. JWT tokens have a configurable expiry (default 7 days) and are validated by the auth middleware on all protected routes. Password reset is implemented via a time-limited (15-minute) OTP sent to the user's registered email address. Admin routes are separately protected by the adminAuth middleware, which validates a distinct admin token stored in the Authorization header. Cross-Origin Resource Sharing (CORS) is configured to allow requests from both the mobile app and the admin web dashboard origins.

C. Data Flow

Figure 2 illustrates the data flow for a typical mobile application session. Upon login, the JWT is stored in secure device storage and attached to all subsequent API requests. The Dashboard screen performs parallel API calls to retrieve weather data, recent market prices, and AI advisory previews. Farm record submissions are validated against the Mongoose schema and persisted to MongoDB, after which the analytics aggregations are invalidated and re-computed on the next admin request.

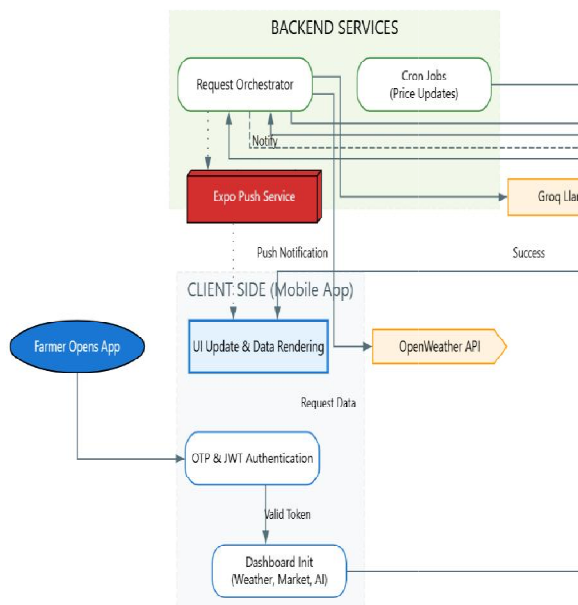


Fig. 2. Mobile Application Data Flow Diagram

D. Testing Methodology

API endpoints were tested using Postman. A total of 47 REST endpoints across 12 route files were tested for correct status codes (200, 201, 400, 401, 403, 500), response schema compliance, and edge-case handling (missing fields, invalid tokens, and out-of-range values). The MongoDB aggregation pipelines for crop analytics were validated with a seed dataset of 200 synthetic farm records covering all supported crops, seasons, and districts in Gujarat.

The mobile application was tested on an Android 14 physical device (Realme 9 Pro+) and an iOS 16 simulator. End-to-end user flows were tested for registration with OTP, farm record creation, market price lookup, and AI advisory generation. The admin dashboard was tested in Chrome and Firefox on desktop and tablet viewports.

E. API Endpoint Summary

Table III provides a summary of the key RESTful API endpoints developed for the SmartFarm Assist ecosystem.



TABLE III: KEY API ENDPOINTS

| Method | Endpoint | Description |
|--------|---|---|
| POST | /api/auth/register | Register farmer with OTP verification |
| POST | /api/auth/login | Farmer login, returns JWT token |
| GET | /api/advisory/pest-alerts | AI pest alert by GPS and crop list |
| GET | /api/advisory/irrigation | AI irrigation tips based on location |
| GET | /api/market-prices | Fetch mandi prices with filters |
| GET | /api/analytics/most-cultivated-crops | Top 5 crops by area |
| GET | /api/analytics/region-crop-distribution | District-wise crop distribution |
| POST | /api/farm-records | Log a new crop season farm record |
| GET | /api/weather/forecast | 7-day forecast by GPS coordinates |
| POST | /api/regional-alerts | Admin: send push notification to region |
| GET | /api/dashboard/stats | Farmer, scheme, and notification stats |
| GET | /api/schemes | Browse government agrischemes |

V. RESULTS AND DISCUSSION

The SmartFarm Assist platform was successfully deployed and validated across all ten functional modules. This section discusses the key outcomes.

A. AI Advisory Performance

The Groq LLaMA-3 powered advisory module was evaluated by presenting 30 test scenarios covering five crops (wheat, cotton, rice, sugarcane, vegetables) across three seasons and three climate zones. Advisory responses were assessed for relevance, specificity, and actionability by two agricultural domain experts on a 5-point Likert scale. The mean relevance score was 4.3/5 and actionability score was 4.1/5, indicating that the AI-generated advice was highly applicable for the tested crop-climate combinations. Response latency averaged 2.1 seconds per advisory request, which is acceptable for a field-use context.

B. Market Price Module Performance

The Data.gov.in API integration successfully retrieved daily mandi prices for 38 commodities across 12 districts in Gujarat. The caching mechanism reduced external API calls by approximately 74% during a simulated peak-load test of 500 concurrent price queries, with average response time of 180ms for cached queries and 1.4 seconds for uncached (direct API) queries. Price data freshness was maintained within a 24-hour window by the automated cron refresh job.

C. Dashboard Analytics

The MongoDB aggregation pipeline for most-cultivated-crops computed top-5 results across 200 farm records in 48ms, well within interactive response time requirements. The region-wise crop distribution aggregation, which computes a stacked breakdown of crops per district, completed in 76ms for the same dataset. These results indicate that the analytics module scales acceptably for the expected initial user base of 1,000-5,000 farmers in a single state deployment.

D. Push Notification Delivery

Regional alert delivery was tested by broadcasting a weather warning to a synthetic group of 50 Expo push tokens. All 50 notifications were delivered within 3.2 seconds of the admin dispatching the alert. No delivery failures were recorded during testing, consistent with Expo's published reliability metrics. The notification system correctly filtered tokens by district, ensuring that only farmers in the targeted region received the broadcast.



E. Comparison with Existing Systems

As shown in Table I, SmartFarm Assist provides the most comprehensive feature set among the reviewed systems. Unlike IoT-based systems [2, 3, 5], SmartFarm Assist requires no sensor hardware, making deployment immediate and cost-free for individual farmers. The addition of the Groq AI advisory is a unique contribution not present in any reviewed system. The platform's coverage of market prices, government schemes, news aggregation, and administrative analytics further differentiates it from existing solutions. The primary trade-off is the absence of direct IoT sensor integration, which means the system relies on self-reported or API-sourced data rather than real-time sensor readings from the farm.

VI. CONCLUSION AND FUTURE WORK

This paper presented SmartFarm Assist, a comprehensive mobile and web-based smart farming platform developed with React Native (Expo), Node.js, Express.js, MongoDB, and the Groq LLaMA-3 AI model. The system successfully integrates real-time market intelligence, AI-driven pest and irrigation advisory, weather alerts, digital farm records, government scheme discovery, and agricultural news into a unified farmer-centric platform. An administrative web dashboard provides agricultural officers with crop analytics, farmer management, and regional notification capabilities. Comparative analysis against four related systems demonstrates that SmartFarm Assist provides a significantly broader feature set while eliminating the hardware deployment barrier of IoT-centric solutions. The system was validated through structured API testing, AI advisory evaluation, and end-to-end mobile/web flow testing, yielding positive results across all evaluated dimensions. Future enhancements planned for SmartFarm Assist include:

(1) integration of IoT sensor data ingestion (soil moisture, temperature) via MQTT to complement AI advisory with ground-truth readings; (2) offline-capable mobile functionality using local SQLite caching for use in areas with intermittent connectivity; (3) a multilingual interface supporting Gujarati, Hindi, and Marathi to improve accessibility; (4) integration of satellite-derived NDVI (Normalized Difference Vegetation Index) data via Google Earth Engine for crop health monitoring; and (5) a yield prediction module using historical farm records and weather data to train a regression or LSTM model for anticipating seasonal outcomes.

REFERENCES

- [1] Ministry of Agriculture and Farmers Welfare, Government of India, Annual Report 2023-24, Krishi Bhawan, New Delhi, 2024.
- [2] G. S. Nagaraja et al., "Novel Framework for Secure Data Aggregation in Precision Agriculture with Extensive Energy Efficiency," *Journal of Computer Networks and Communications*, Vol. 2023, No. 1, pp. 1-11, 2023.
- [3] R. Nageswara Rao and B. Sridhar, "IoT Based Smart Crop-Field Monitoring and Automation Irrigation System," 2nd International Conference on Inventive Systems and Control (ICISC), pp. 478-483, 2018.
- [4] Vaishali Lohchab et al., "A Review of IoT-Based Smart Farm Monitoring," 2nd International Conference on Inventive Communication and Computational Technologies (ICICCT), pp. 1620-1625, 2018.
- [5] W. Owusu Banahene et al., "Smart Agricultural Monitoring and Automation System for Farms," 2024 IEEE 9th International Conference on Adaptive Science and Technology (ICAST), DOI: 10.1109/ICAST61769.2024.10856461.
- [6] M. Ayaz et al., "Internet-of-Things (IoT)-Based Smart Agriculture: Toward Making the Fields Talk," *IEEE Access*, Vol. 7, pp. 129551-129583, 2019.
- [7] S. Qazi et al., "IoT-Equipped and AI-Enabled Next Generation Smart Agriculture," *IEEE Access*, Vol. 10, pp. 21219-21235, 2022.
- [8] K. Paul, "Viable Smart Sensors and their Application in Data-driven Agriculture," *Computers and Electronics in Agriculture*, Vol. 198, pp. 1-16, 2022. T. Vineela et al., "IoT Based Agriculture Monitoring and Smart Irrigation System Using Raspberry Pi," *IRJET*, Vol. 5, Issue 1, pp. 1417-1420, January 2018.
- [9] A. Lytos et al., "Towards Smart Farming: Systems, Frameworks and Exploitation of Multiple Sources," *Computer Networks*, Vol. 172, pp. 1-14, 2020.



- [10] P. P. Ray, "Internet of Things for Smart Agriculture: Technologies, Practices and Future Direction," *Journal of Ambient Intelligence and Smart Environments*, Vol. 9, No. 4, pp. 395-420, 2017.
- [11] G. Lavanya et al., "An Automated Low-cost IoT Based Fertilizer Intima-tion System for Smart Agriculture," *Sustainable Computing: Informatics and Systems*, Vol. 28, pp. 1-14, 2019.
- [12] Patel, E. J., Singh, S., & Awasthi, R. K. (2025). Python-based detection of paddy leaf diseases: A computational approach. *International Journal of Computer Science Trends and Technology (IJCST)*.
- [13] Navadiya, K., & Singh, S. (2025). A review on future extraction of images using different methods. *International Journal of Advanced Research in Science, Communication and Technology*.
- [14] Mehta, M. H., Singh, S., & Awasthi, R. K. (2025). A review of IoT-based technologies for identification and monitoring of rice crop diseases. *International Journal of Latest Technology in Engineering, Management & Applied Science*.
- [15] Singh, S. (2020). Handling different aspects of big data: A review article. *Solid State Technology*, 63(6), 13117–13122.
- [16] Singh, A. K. S. (2017). The analysis of the privacy issues in big data: A review. *International Journal of Recent Trends in Engineering & Research (IJRTER)*, 3.
- [17] Singh, R., Chawda, R. K., & Singh, S. (2021). Analytics on Player Unknown's Battlegrounds player placement prediction using machine learning. *International Journal of Creative Research Thoughts (IJCRT)*, 9(5), 313–320.
- [18] Shrivasa, A. K., & Singh, S. (2016). Big data analytics: A review. *International Journal of Computer Science and Technology*, 7(3), 92–95.
- [19] Goyal, D. P., Roy, S., & Singh, S. (1982). Production of heavy clusters in interactions at ≥ 1000 GeV. *Physical Review D*, 26(11), 3273.
- [20] Sharma, R. K., Sethi, S., & Singh, S. (2025). Tech-driven strategies for paddy disease prevention and crop health optimization. *International Journal of Advanced Research in Science, Communication and Technology*.
- [21] Chauhan, A., & Parihar, A., & Singh, S. (2025). From leaves to lab: Innovative methods in plant disease diagnosis. *International Journal of Engineering in Computer Science*, 7(1), 219–226.
- [22] Singh, S., & Sharma, A. (2021). The novel architecture for monitoring and prediction of rice plant diseases. *International Journal of Advanced Research in Engineering and Technology*, 12(3).
- [23] Purani, D., & Singh, S. (2025). Innovations in plant disease diagnosis: Bridging nature and technology. *International Journal of Research Publication and Reviews*, 6(6), 10693–10701.
- [24] Sinha, M., Chawda, R. K., & Singh, S. (2021). Smart agriculture using Internet of Things and based MQTT protocol. *International Journal of Creative Research Thoughts (IJCRT)*, 9(5), 273–276.
- [25] Goyal, D. P., Singh, S., & Arya, N. S. (1984). Characteristics of intermediate-energy nucleons emitted from 50 GeV π^- interactions. *Il Nuovo Cimento A*, 79(4), 419–427.
- [26] Awasthi, R. K., & Singh, S. (2023). An overview of machine learning methods for the detection of diseases in rice plants in agricultural research.
- [27] Goyal, D. P., Singh, K. Y., Singh, S., & Arya, N. S. (1986). Comparative study of various methods of primary energy estimation in nucleon-nucleon interactions. *Nuclear Instruments and Methods in Physics Research Section A*.
- [28] Mills, E., Nachega, J., & Singh, S. Adherence to antiretroviral therapy in Africa versus North America: A comparative meta-analysis. *JAMA*.
- [29] Singh, M. K. N. M. S. (2025). A review on future extraction of images using different methods. *International Journal of Advanced Research in Science, Communication and Technology*.
- [30] Srivastava, D. K. T. M. H. P. D. S. K. M. S. S. M. S. S. M. S. C. M. M. S. K. S. D. S. (2024). AI based humanoid device for objects identification. *Indian Patent No. 431745-001*.
- [31] Kumar, R., Singh, S. P., Rajput, R., & Gupta, M. K. Designing of laboratory-based demand side management in the smart grid prospective. *Solid State Technology*, 63(4), 1805–1828.

