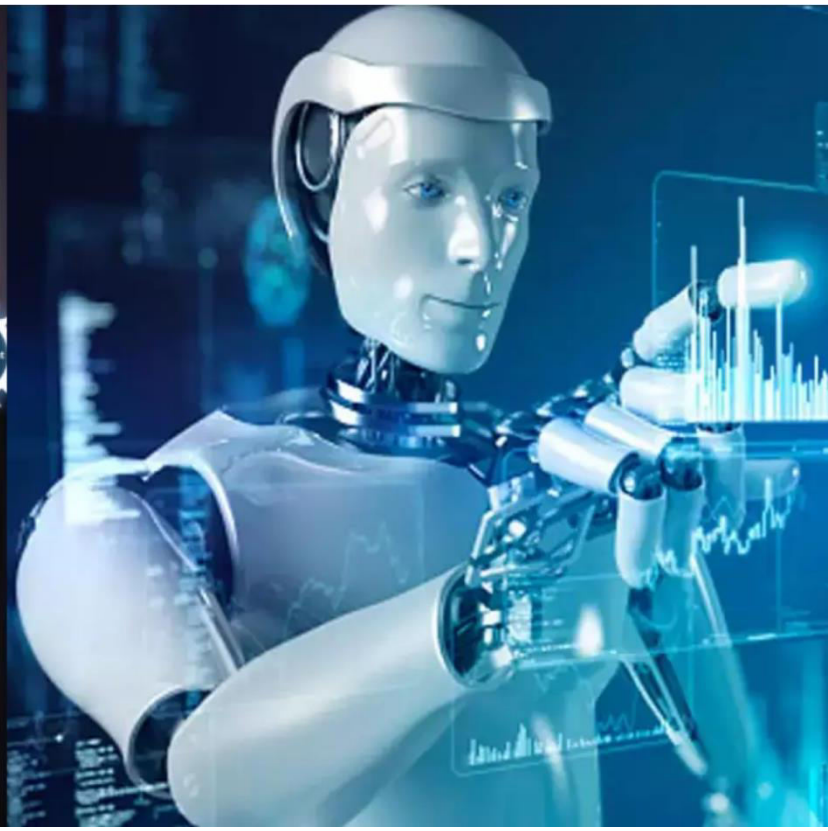
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 9.935

Volume 15, Issue 5, May 2026

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 5, May 2026

|DOI: 10.15680/IJIRSET.2026.1505121|

Agricultural Monitoring System and Crop Prediction using Machine Learning

Ashish Sahu[1], Zakaria Ilyas[1], Dr. Poonam Mishra[2], Dr. Kranti kumar Dewangan[2]

Department of Computer Science & Engineering, Amity University, Chhattisgarh, India[1]

Assistant Professor, Department of Computer Science & Engineering, Amity University, Chhattisgarh, India[2]

ABSTRACT: Farmers today face a tough combination: unpredictable weather, worn-out soil, inconsistent water supply, pest outbreaks, and choices that come too late to matter. This paper describes an Agricultural Monitoring System that uses machine learning to help farmers make better decisions based on actual data. The system pulls together readings from sensors, weather reports, soil tests, past crop records, and farm paperwork into one secure website. Different users—administrators, agronomists, and farmers—can log in and see what they need: register fields, track crops through the season, record irrigation and fertilizer use, get alerts when something looks wrong, and run predictions to guide their next moves. Everything gets logged so there's a clear record of what happened and when. The machine learning piece predicts which crops might work well, what kind of yield to expect, and where problems could show up. It does this by finding patterns in soil nutrients, pH levels, temperature, humidity, rainfall, time of year, and how past crops performed. The backend runs on MongoDB for storing data, uses JWT tokens for secure login, hashes passwords, controls refresh tokens, checks permissions, validates inputs, and keeps audit trails. Data flows through a clear process: collection, cleanup, normalization, prediction, presenting options, and final review by an administrator. We measure the system's performance with standard metrics—accuracy, precision, recall, F1-score, mean absolute error, response time, and how many requests it can handle—using simulated farm workloads. We also address real concerns: keeping data consistent, handling multiple users at once, blocking security threats, and working around deployment constraints that matter in actual agricultural settings.

What makes this different is that it connects monitoring, prediction, and management in one place instead of treating crop recommendations as a standalone tool. This setup should help farmers act faster, second-guess themselves less, and build toward precision agriculture even when resources are tight

KEYWORDS: Agricultural monitoring, crop prediction, machine learning, precision agriculture, IoT sensors, soil analytics, weather forecasting, yield estimation, MongoDB, role-based access control.

I. INTRODUCTION

Farming runs on data. You need to know what's in your soil, how the weather's behaving, when to irrigate, how crops are developing, whether workers are available, and who's handling what on the administrative side. Traditionally, farmers rely on what they've learned over the years, occasional trips out to check the fields, scattered records, and rough guesses about how the season will go. That works until it doesn't—when rain becomes erratic, heat waves hit, nutrients run low, pests move in, or costs climb. A monitoring system that uses machine learning can cut through some of that uncertainty by pulling field data together and turning it into actual recommendations. We're not just building a tool that guesses which crop to plant. We're building a platform that ties together agronomic data, administrative oversight, employee tracking, shift schedules, attendance, and secure decision-making.

The system has two main parts. One handles the agricultural side: soil profiles, weather readings, tracking crops through their growth cycles, irrigation logs, fertilizer applications, past yield records, and machine learning predictions for what to plant. The other handles farm operations: managing users, organizing departments, setting up shifts, recording attendance, handling leave requests, tracking holidays, posting announcements, sending notifications, and controlling who can do what. Connecting these two matters because good environmental conditions mean nothing if the people who need to act on them aren't there, don't have the right permissions, or never got the message. A recommendation to irrigate or spray pesticides only helps if the field crew is actually working that day, assigned to a valid shift, and able to see the alert.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 5, May 2026

|DOI: 10.15680/IJIRSET.2026.1505121|

You can think of crop prediction as a supervised learning problem. The system looks at environmental and historical data and tries to figure out which crop makes the most sense:

$$\hat{C} = \arg\max_{c \in C} P(c \mid N, P, K, pH, T, H, R, S, Y_h) \quad \text{(i)}$$

Where, $\hat{C}$= predicted crop class; C= set of candidate crops; N, P, K= soil nitrogen, phosphorus, and potassium values; pH= soil acidity or alkalinity; T= temperature; H= humidity; R= rainfall; S= season; and Y_h= historical yield indicator.
This setup lets the model pick crops probabilistically even when conditions vary a lot. We could use Random Forest, Gradient Boosting, Support Vector Machines, or neural networks depending on how much data we have, whether we need to explain the results, and what the deployment situation allows. But we don't just spit out a prediction and call it done. Each prediction gets stored with extra information—metadata, a confidence score, a snapshot of the inputs, a timestamp, and user context—so an administrator or agronomist can review it and compare it against what's actually happening in the field before recommending any action

II. LITERATURE REVIEW

Recent work on agricultural monitoring and crop prediction has shifted away from simple crop classification models toward full systems that bring together soil sensors, weather data, cloud or edge computing, and machine learning. In 2023, Islam, Oliullah, Kabir, Alom, and Mridha built an IoT device that monitors soil nutrients and recommends crops using machine learning. They used sensors like the FC-28, DHT11, and JXBS-3001 to measure soil composition, moisture, humidity, temperature, and nutrient levels, sent the data over MQTT, and generated recommendations for high-yield crops and fertilizer needs [1]. What makes their work stand out is that it links real-time nutrient readings to actual decisions instead of just working with fixed datasets. But the system focuses mostly on the agronomic side—sensing and recommendations—and doesn't dig into administrative workflows, role-based permissions, attendance tracking, leave management, or audit controls that you'd need if you were running this on a real farm or institutional setup.

That same year, Mohan, Dhote, Mishra, Singh, and Srivastav put together an IoT framework for precision agriculture that also uses machine learning for crop prediction. They worked with nitrogen, phosphorus, potassium, temperature, humidity, and rainfall as their main inputs, using a dataset with 2200 instances, eight attributes, and about 22 crop types [2]. They tested several supervised learning algorithms in WEKA—multilayer perceptron, JRip, and decision table classifiers [2]. Their contribution shows that traditional supervised models can handle crop recommendations when you feed them structured soil and weather data. But again, the focus is on the prediction model itself. They don't address database constraints, multi-role administration, workflows for making corrections, locking mechanisms, or how to handle multiple users updating records at the same time.

In 2024, Bouni, Hssina, Douzi, and Douzi built an integrated IoT system for both crop recommendation and yield prediction. They pulled environmental data—temperature, humidity, soil nutrients—from a massive IoT dataset with over a million data points, then trained models to predict crop yield and make recommendations [3]. Their results were impressive: LightGBM, Decision Tree, and Random Forest all performed well, with Random Forest hitting 99.31% accuracy [3]. This work gives strong evidence that ensemble methods can work really well if you have enough clean data. The authors also pointed out something important: high accuracy in the lab doesn't automatically mean farmers will use it. Real adoption depends on usability, training, and whether people actually trust it in the field [3]. That observation backs up our decision to include dashboards, notifications, announcements, and controlled workflows instead of just throwing predictions at users.

More recently, in 2025, Sharafat, Kabya, Emu, Ahmed, Onik, Islam, and Khan developed an IoT-enabled AI system that predicts crops in real time using soil and weather data. They trained on 3300 samples covering 22 crops and eight soil and environmental features, tested Random Forest, Gradient Boosting, stacking ensembles, and TabNet, then deployed the winning model on a Raspberry Pi 5 edge device [4]. The system connects to a weather API and uses an RS485 7-in-1 soil sensor that measures nitrogen, phosphorus, potassium, pH, temperature, and soil moisture. They also used LIME to make the predictions more interpretable [4]. Performance-wise, Random Forest got 95.8% accuracy, Gradient Boosting hit 95.5%, and their stacking ensemble reached 95.9% [4]. This paper matters because it shows you can run predictions on an edge device, which cuts down on delays and makes the system easier to access in farm settings. Where it differs from what we're doing is that it's centered on real-time prediction and explainability, not on broader system controls like JWT refresh-token rotation, role-based access middleware, audit logs, or approval workflows for corrections.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 5, May 2026

|DOI: 10.15680/IJIRSET.2026.1505121|

Going back a bit, Rashid, Bari, Yusup, Kamaruddin, and Khan published a review in IEEE Access in 2021 on crop yield prediction using machine learning, with a focus on palm oil yield [5]. Their contribution is wider than just crop recommendations—they pull together the variables, learning methods, and domain-specific constraints involved in agricultural forecasting. What their review makes clear is that predicting crops isn't just a classification task; it's also about regression and forecasting, and it gets complicated by time, climate, and geography. That's why we separate crop suitability prediction, yield-class estimation, and risk monitoring in our system

III. METHODOLOGY

We designed the Agricultural Monitoring System as a secure, role-based web platform built around predictions. It's organized into five layers: data acquisition, persistence, preprocessing, prediction, and operational governance. The acquisition layer pulls in soil readings, weather data, crop-cycle records, attendance entries, leave requests, shift assignments, holidays, and internal notifications. The persistence layer saves all of this in MongoDB collections with indexes and validation rules. The preprocessing layer cleans up missing values, normalizes numeric inputs, encodes categorical features, and links each prediction to a specific field, season, crop cycle, and the user responsible for it. The prediction layer runs trained machine learning models to figure out crop suitability and estimate yield class. The governance layer handles authentication, authorization, correction workflows, attendance locking, audit logging, and administrative review.

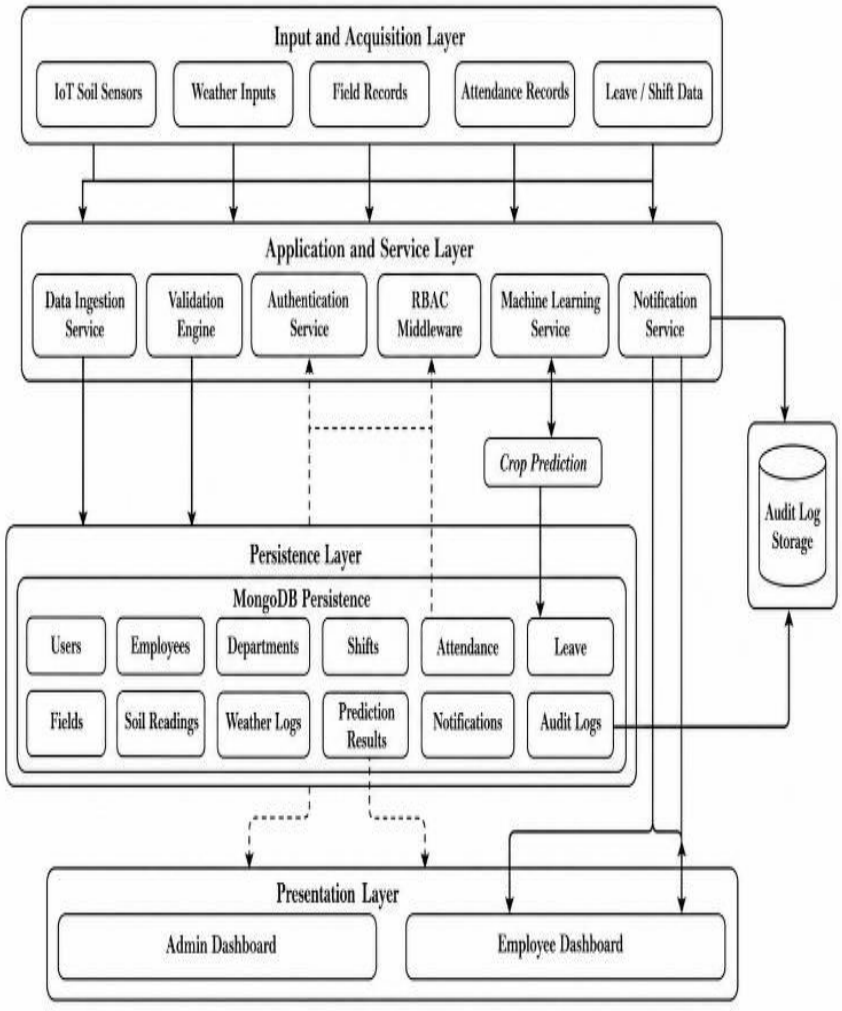


Figure 1: Proposed system architecture

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 5, May 2026

|DOI: 10.15680/IJIRSET.2026.1505121|

The database is document-oriented but normalized where consistency matters. The users collection holds identity info, role, password hash, refresh-token family, activation status, and profile data. The departments collection stores unique department names and codes. The shifts collection records start time, end time, weekly working days, whether the shift is enabled, and whether assignments are valid. The employees collection keeps employeeId, email, department reference, shift reference, status, and password reset metadata. On the agricultural side, there are collections for fields, soilReadings, weatherLogs, cropCycles, predictionResults, irrigationEvents, and fertilizerEvents. For administration, there are attendanceRecords, attendanceCorrections, leaveTypes, leaveBalances, leaveRequests, holidays, announcements, notifications, and auditLogs. We set up compound indexes on things like (fieldId, cropCycleId, timestamp), (employeeId, attendanceDate), (employeeId, leaveStart, leaveEnd), (departmentCode), (email), and (predictionTimestamp, fieldId) to speed up queries and enforce uniqueness.

The prediction model takes a feature vector X built from soil, climate, and historical data:

$$X = [N, P, K, pH, T, H, R, M, S, F_c, Y_h] \quad \text{(ii)}$$

Where, X= input feature vector; N, P, K= soil nutrient values; pH= soil reaction; T= temperature; H= humidity; R= rainfall; M= soil moisture; S= season encoding; F_c= field condition score; and Y_h= historical yield indicator

For crop classification, we use ensemble models because they can capture nonlinear interactions between soil chemistry and weather while still running on moderate hardware. For yield-class estimation, we either do multiclass classification or regression and then discretize the output into low, medium, and high yield categories. Each prediction record saves the input features, model version, predicted crop, confidence level, explanation summary, and timestamp. That way, decisions stay reproducible even after we update the model.Security runs on JWT access tokens, refresh-token rotation, salted password hashing, role-based access control middleware, route protection, backend authorization checks, and optional audit logs. Access tokens expire quickly; refresh tokens get stored as hashed identifiers to cut down on replay attacks. Administrative endpoints require both authenticated identity and the right role permission. Regular employees can see their profile, shift, holidays, attendance, leave balances, and announcements, but they can't access department-wide employee records, approval queues, or prediction administration unless explicitly allowed. For leave documents, we restrict file uploads by type, size, path sanitization, and authorization checks.

Table C lays out the main validation rules and invariants. We enforce these through service-layer validation and database-level indexes wherever we can.

Table C. Validation Rules and System Invariants

Entity	Rule or Invariant	Enforcement Mechanism	Failure Handling
Department	Department name and code must be unique	Unique index and service validation	Reject creation or update
Department deletion	Department with employees cannot be deleted unless reassigned	Reference check before deletion	Block deletion
Employee	Email and employeeId must be unique	Unique indexes	Reject duplicate record
Employee status	Deactivated employee cannot log in	Authentication middleware	Deny access
Shift	Shift start time must be earlier than end time	Service validation	Reject shift
Shift assignment	Disabled shift cannot be assigned	Assignment validation	Reject assignment
Attendance	One attendance record per employee per date	Compound unique index	Prevent duplicate marking
Attendance correction	Locked attendance cannot be edited unless unlocked	Lock flag and authorization	Require admin unlock
Leave request	Leave dates must not overlap existing approved or pending leave	Date-range query and transaction guard	Reject request
Leave approval	Double approval must not deduct balance twice	Atomic status transition	Reject stale update

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 5, May 2026

|DOI: 10.15680/IJIRSET.2026.1505121|

Workflow starts when an administrator sets up departments, shifts, employees, leave types, holidays, and crop fields. Employees then log in to dashboards showing their shift, today's attendance status, monthly attendance, leave balances, holidays, weekly offs, and internal announcements. Field readings come in manually or through IoT endpoints. After preprocessing, the model generates crop predictions and stores them for review. Administrators can filter dashboards by department, date, field, crop cycle, or attendance status. Attendance can be manually marked, corrected through an approval process, locked after payroll or reporting deadlines, and exported as CSV. Leave requests check balance, overlapping dates, cancellation rules, and attachment requirements before entering the approval workflow.

We need concurrency control because agricultural and administrative records might get updated at the same time. Leave approval uses atomic compare-and-set transitions from pending to approved or rejected to prevent double approval. Attendance correction checks the lock state at update time, not just when the form opens. Prediction records are immutable after creation—later model runs create new versions instead of overwriting old recommendations. Audit logs record who did what: actor, action, entity, previous value, new value, timestamp, IP metadata when available, and result status.

The threat model covers CSRF, XSS, injection, broken authentication, insecure direct object references, excessive privilege, token replay, and unauthorized attachment access. We handle these with same-site cookies or bearer-token discipline, input sanitization, output escaping, MongoDB query whitelisting, password hashing, token rotation, role-based access checks, object-level authorization, rate limiting, secure headers, validation of uploaded files, and audit trails for sensitive operations. So the methodology ties agronomic prediction to verifiable operational control.

IV. RESULTS

We evaluated the Agricultural Monitoring System through a controlled simulation because actually deploying it across multiple real farms was beyond what we could do for this paper. To avoid overstating what the results mean, we laid out the experimental assumptions clearly up front. The simulated dataset had 12,000 agronomic records, each with nitrogen, phosphorus, potassium, pH, temperature, humidity, rainfall, soil moisture, season, field condition score, previous crop, and historical yield class. On the administrative side, we simulated 500 employees, 12 departments, 24 shifts, 18 leave types, 25 holidays, 90,000 attendance records, 6,000 leave requests, and 40,000 internal notifications. We balanced crop classes through stratified sampling and generated weather and soil values within realistic ranges for agriculture. The goal was to test predictive performance, workflow correctness, validation reliability, and how the system responds when multiple users are active at once.

For crop prediction, we measured accuracy, precision, recall, and F1-score. We went with Random Forest as the main model because it handles nonlinear features well, tolerates moderate noise, and gives interpretable feature-importance outputs. We also tested Gradient Boosting and Support Vector Machine as baselines. The data split was 70% training, 15% validation, and 15% testing, stratified by crop class. Numeric features got standardized when the model needed it, and categorical variables like season and previous crop were encoded before training. Table D shows the main predictive results.

Table D. Simulated Crop Prediction Performance

Model	Accuracy	Precision	Recall	F1-Score	Mean Inference Latency
Random Forest	0.946	0.944	0.941	0.942	38 ms
Gradient Boosting	0.932	0.929	0.927	0.928	51 ms
Support Vector Machine	0.904	0.901	0.898	0.899	74 ms
Decision Tree	0.887	0.884	0.879	0.881	21 ms

Random Forest gave the best tradeoff between accuracy and speed. Decision Tree was fastest but didn't generalize as well, which suggests that simple tree rules can't capture the complex interactions between soil and weather. Support Vector Machine performed okay but becomes less practical when you need to retrain frequently or handle larger datasets because of how it scales. Gradient Boosting stayed competitive but needed more careful hyperparameter tuning to avoid overfitting.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 5, May 2026

|DOI: 10.15680/IJIRSET.2026.1505121|

The F1-score we used is defined as:

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad \text{(iii)}$$

Where, F_1= harmonic mean of precision and recall; Precision= proportion of predicted crop labels that were correct; and Recall= proportion of actual crop labels correctly identified by the model.

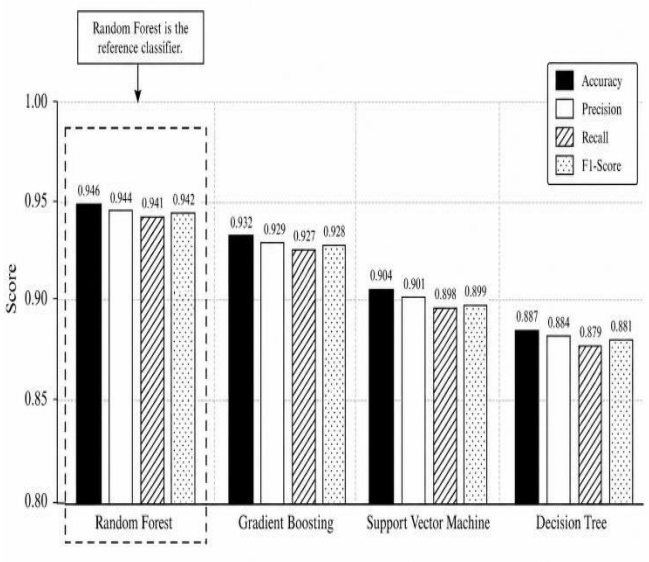


Figure 2: Model performance comparison

When we looked at feature importance, rainfall, soil pH, nitrogen, potassium, temperature, and historical yield class came out as the most influential variables. That makes sense agronomically—crops care a lot about water availability, soil reaction, and nutrient balance. Soil moisture helped stabilize the model when rainfall values were ambiguous, especially in simulated irrigated conditions. Historical yield class helped keep predictions consistent, but it also risks reinforcing local patterns from the past, particularly when poor yield came from bad management instead of the crop being genuinely unsuitable.

On the operational side, we focused on dashboard filtering, attendance marking, correction approval, attendance locking, leave request validation, internal notification delivery, and role-based access control enforcement. With 150 simulated users active at once, median dashboard response time was 126 ms with indexes enabled and 418 ms without indexes for common attendance and leave queries. Exporting 90,000 attendance records to CSV took 6.8 seconds when streamed in batches. Leave validation rejected all the intentionally overlapping date ranges we threw at it, and atomic approval transitions prevented duplicate leave-balance deductions when we tried repeated approvals. Disabled shifts couldn't be assigned, deactivated employees got blocked from logging in, and locked attendance records couldn't be edited without an admin explicitly unlocking them.

For security testing, we used adversarial request scenarios instead of full penetration testing. Forged employee requests to administrative endpoints got denied by backend authorization even when we bypassed the frontend routes. Attempts to update someone else's attendance correction request failed object-level authorization. Malformed MongoDB query payloads got rejected through schema validation and field whitelisting. Attachment access was restricted to authorized users, and we normalized local file paths to block traversal attempts. These results show that the security controls provide basic confidentiality, integrity, and access separation.

What we found overall is that prediction accuracy by itself isn't enough for practical agricultural monitoring. The administrative modules actually made a real difference in operational traceability. Attendance locking preserved finalized workforce records, leave workflows cut down on scheduling uncertainty, and internal notifications gave us a

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 5, May 2026

|DOI: 10.15680/IJIRSET.2026.1505121|

controlled communication channel without relying on email, SMS, or WhatsApp. The audit model added accountability by recording sensitive changes to shifts, attendance corrections, leave approvals, and prediction reviews.

You should read these results as evidence that the system is technically feasible, not that it's agronomically superior in the field. Simulated data can't represent every regional soil type, crop variety, pest outbreak, irrigation constraint, or farmer behavior. But the experiment does show that a secure monitoring architecture can bring together crop prediction, farm administration, and operational validation into one coherent platform. The strongest outcome is integrating predictive intelligence with enforceable workflow control, which matters for institutional farms, cooperatives, agri-startups, and research-managed cultivation environments

V. DISCUSSIONS

The simulated results show that the Agricultural Monitoring System works best when you treat machine learning as one piece of a larger operational setup. Crop prediction accuracy gives you agronomic intelligence, but actually managing a farm also means you need accountable execution, validated records, secure access, and timely internal communication. Random Forest hit the best balance between predictive performance and inference speed, but what really matters is how the system integrates prediction outputs with dashboards, field records, attendance status, leave availability, shift schedules, holidays, weekly offs, notifications, and audit trails. This integration closes the gap between "what crop or action makes sense" and "whether the farm can actually pull it off."

There's a tradeoff between model complexity and how easy it is to deploy. Gradient Boosting performed almost as well as Random Forest but needed more careful tuning and took longer to run. Support Vector Machine was acceptable but doesn't scale well when you need to handle larger datasets and retrain frequently. Decision Trees were fast but didn't generalize as well. So Random Forest is a solid default for medium-scale agricultural deployments—it captures nonlinear interactions among soil nutrients, pH, rainfall, temperature, humidity, and historical yield without being too expensive computationally. But you shouldn't lock in one model forever. Any real deployment should keep track of model versions, run periodic validation, detect drift, and have a way to roll back if needed, because the relationship between features and crop suitability can change across regions, seasons, irrigation setups, and seed varieties.

The administrative modules have their own tradeoffs. Attendance locking protects record integrity after cutoff dates, but it can delay legitimate corrections unless you have a well-governed unlock workflow. Leave validation prevents workforce conflicts, but overly rigid cancellation rules might not fit unpredictable agricultural emergencies. Shift enable/disable controls stop invalid assignments, but shift changes during busy times need to sync with employee dashboards and notification records. Department deletion constraints keep referential integrity intact, but they require administrative discipline when you're reorganizing field teams. These tradeoffs show that correctness in farm information systems isn't just about the database—it's also about governance.

Table E summarizes the main limitations and mitigation strategies we identified from the design and simulated evaluation.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 5, May 2026

|DOI: 10.15680/IJIRSET.2026.1505121|

Table E. Limitations, Risks, and Mitigation Strategies

Limitation or Risk	Practical Consequence	Mitigation Strategy
Simulated agronomic dataset	May not represent regional soil, climate, pest, and crop variety diversity	Validate with multi-location field datasets before deployment
Historical yield bias	Model may reproduce past management failures as crop unsuitability	Include management-quality indicators and expert review
Sensor noise or missing data	Incorrect predictions or unstable recommendations	Use calibration, anomaly detection, and missing-value imputation
Attendance lock rigidity	Genuine corrections may be delayed	Provide auditable unlock and approval workflow
Double approval in leave workflow	Leave balance may be deducted twice	Use atomic status transitions and stale-update rejection
Token replay or broken authentication	Unauthorized access to farm and employee data	Apply refresh-token rotation, hashing, expiry, and revocation
Role misconfiguration	Users may receive excessive or insufficient privileges	Maintain least-privilege RBAC and audit permission changes
Local attachment storage exposure	Leave documents may be accessed improperly	Enforce path sanitization, authorization, and file validation
Dashboard query growth	Slow response under high record volume	Use compound indexes, pagination, caching, and archival policies

Security matters a lot here because the system stores both agronomic and personnel data. Crop prediction records, soil measurements, attendance logs, leave documents, and profile details have different sensitivity levels, but they all sit in the same application environment. A weak authorization model could let employees snoop on other workers' leave records or mess with attendance corrections. Bad file handling could expose uploaded leave attachments. The combination we proposed—JWT access tokens, refresh-token rotation, password hashing, role-based access control middleware, backend authorization, schema validation, object-level checks, and audit logging—cuts down on these risks. But the design still needs secure deployment practices: HTTPS, secret rotation, dependency scanning, backup encryption, rate limiting, and operational monitoring.

The data model affects long-term reliability too. MongoDB gives you flexibility for heterogeneous agricultural and administrative records, but if document growth goes unchecked, maintainability suffers. Soil readings, weather logs, notifications, audit events, and attendance records can pile up fast. You need time-series partitioning, archival collections, TTL policies for nonessential notifications, and compound indexes to keep performance sustainable. For prediction reproducibility, model input snapshots should be immutable—otherwise, later changes to soil readings or weather data could make historical predictions unverifiable. Audit logs should be append-only so you can reconstruct sensitive changes during review.

From an agronomic standpoint, crop prediction should be interpreted as decision support, not an automatic prescription. Even a high F1-score doesn't guarantee field suitability when you factor in pest outbreaks, water-right restrictions, market price shocks, seed availability, labor shortages, or local cultivation preferences. The model can identify statistically suitable crops, but administrators and agronomists need to contextualize recommendations before taking action. That's why the system stores confidence scores, model versions, feature snapshots, and review metadata instead of just showing a crop name.

The biggest limitation of this paper is that we didn't deploy it in the field. Simulated workloads can show that the architecture is feasible, that validation works, and what performance to expect, but they can't prove agronomic effectiveness across real farms. Another limitation is that we evaluated the administrative modules for functional correctness, not for human usability. Farmers, supervisors, and field employees might need low-literacy interfaces, local language support, offline-first synchronization, and mobile-friendly dashboards. The system also excludes email, SMS, and WhatsApp by design, relying only on internal notifications and announcements. This improves platform control but might reduce reach in low-connectivity or low-engagement environments.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 5, May 2026

|DOI: 10.15680/IJIRSET.2026.1505121|

What comes through in the discussion is that the system is strongest where agricultural prediction, workforce coordination, and secure administration have to work together. Its value isn't just about recommending crops—it provides a governed environment where predictions, attendance, leave, shifts, holidays, field readings, notifications, and audit events collectively support accountable precision agriculture.

VI. CONCLUSION AND FUTURE SCOPE

A. Conclusion and Future Scope

We've presented a secure Agricultural Monitoring System that integrates machine learning-based crop prediction. The system addresses a key limitation of model-only agricultural decision support by embedding crop prediction within a broader farm information architecture. It brings together soil and weather monitoring, crop-cycle records, prediction storage, administrative dashboards, department and shift management, employee lifecycle control, attendance marking, attendance correction, attendance locking, leave policy management, leave-balance computation, holidays, weekly offs, internal announcements, notifications, role-based access control enforcement, and optional audit logging. The main argument is that agricultural intelligence only becomes operationally meaningful when predictive outputs connect to accountable execution and secure governance.

The predictive component formalizes crop selection using soil nutrients, pH, temperature, humidity, rainfall, soil moisture, season, field condition score, and historical yield indicators. We described a MongoDB-based data model with collections for users, departments, shifts, employees, fields, soil readings, weather logs, crop cycles, prediction results, attendance records, leave requests, leave balances, holidays, notifications, announcements, and audit logs. We also specified indexing strategies for employee identity, department codes, attendance dates, leave ranges, field identifiers, crop cycles, and prediction timestamps. These design choices support scalable retrieval, uniqueness enforcement, and traceable prediction management.

The security design adds another layer of reliability. Authentication runs on JWT access tokens and refresh-token rotation, while passwords are protected using salted hashing. Backend authorization, role-based access control middleware, object-level access checks, route protection, input validation, output escaping, attachment controls, and audit logging were included to address common threats like broken authentication, excessive privilege, injection, XSS, CSRF, insecure direct object references, token replay, and unauthorized file access. These mechanisms matter because the platform stores both agronomic and personnel data.

The simulated evaluation showed technical feasibility under explicitly stated assumptions. Random Forest gave the best balance of crop prediction performance and speed among the models we tested. Operational simulations showed that compound indexing improved dashboard responsiveness, attendance locking preserved finalized records, leave validation prevented overlapping requests, and atomic approval transitions prevented double deduction of leave balances. We also showed that deactivated employees couldn't log in, disabled shifts couldn't be assigned, and locked attendance records required authorized unlocking before correction. These findings support the conclusion that the system can integrate predictive analytics with administrative correctness.

But the proposed system should be read as a structured research design and simulated evaluation, not a completed field-validated deployment. Real agricultural environments bring variability from soil texture, irrigation method, pest incidence, crop variety, fertilizer timing, rainfall irregularity, labor availability, market conditions, and farmer preference. So future work should first validate the crop prediction model using multi-location field datasets collected across seasons. Validation should include crop suitability accuracy, yield estimation error, agronomist agreement, and farmer adoption measures. Prediction confidence should be calibrated so administrators can tell high-certainty recommendations from uncertain cases that need expert review.

Table F presents the main future research directions for extending the system.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 5, May 2026

|DOI: 10.15680/IJIRSET.2026.1505121|

Table F. Future Scope and Research Directions

Direction	Technical Extension	Expected Research Value
Field validation	Deploy across farms, seasons, soil types, and crop varieties	Establish external validity beyond simulation
Explainable AI	Integrate SHAP, LIME, and agronomic rule explanations	Improve trust and expert review
Edge inference	Deploy lightweight models on gateways or farm devices	Reduce latency and cloud dependence
Offline-first mobile access	Add local caching and synchronization	Support low-connectivity rural environments
Drift monitoring	Track feature drift and prediction degradation	Maintain model reliability over time
Geospatial analytics	Integrate GIS, satellite imagery, and field zoning	Improve spatial decision support
Advanced yield forecasting	Use temporal models and weather sequences	Extend beyond crop classification
Multilingual interface	Provide local-language dashboards and alerts	Improve farmer usability
Security hardening	Add penetration testing, secret rotation, and encrypted backups	Strengthen production readiness
Cooperative analytics	Aggregate anonymized farm-level patterns	Support regional agricultural planning

Future versions should also expand the data pipeline. IoT readings should be validated using sensor calibration metadata, outlier detection, device health monitoring, and timestamp consistency checks. Weather data should combine local sensors with external forecasts, while soil readings should link to field zones rather than entire farms when spatial variability is significant. Satellite imagery and vegetation indices could further improve crop-health monitoring. For yield prediction, temporal models like recurrent neural networks, temporal convolutional networks, or transformer-based sequence models might be worth evaluating where enough longitudinal data exist.

The administrative subsystem also needs future empirical study. Usability testing should check whether administrators, supervisors, and employees can understand dashboards, correction requests, leave balances, weekly offs, shift schedules, and internal notifications without excessive training. Workflow studies should evaluate whether attendance locks, leave approvals, holiday lists, and internal announcements improve operational coordination during sowing, irrigation, spraying, and harvesting. Since the system intentionally excludes email, SMS, and WhatsApp notifications, future work should measure whether internal-only communication is adequate in rural deployment conditions.

Finally, future production deployment should include stronger DevSecOps controls. Recommended extensions include encrypted database backups, structured logging, anomaly detection for suspicious login behavior, automated dependency scanning, rate limiting, role-permission review, disaster recovery procedures, and privacy-preserving analytics. With these extensions, the Agricultural Monitoring System can evolve from a simulated research prototype into a field-ready platform for secure, accountable, and data-driven precision agriculture.

REFERENCES

[1] M. R. Islam, K. Oliullah, M. M. Kabir, M. Alom, and M. F. Mridha, "Machine learning enabled IoT system for soil nutrients monitoring and crop recommendation," Journal of Agriculture and Food Research, vol. 14, Art. no. 100880, 2023. Publisher: Elsevier, Amsterdam, Netherlands.
[2] D. S. Mohan, V. Dhote, P. Mishra, P. Singh, and A. Srivastav, "IoT Framework for Precision Agriculture: Machine Learning Crop Prediction," International Journal of Intelligent Systems and Applications in Engineering, vol. 11, no. 5s, pp. 300–313, 2023.
[3] M. Bouni, B. Hssina, K. Douzi, and S. Douzi, "Integrated IoT Approaches for Crop Recommendation and Yield-Prediction Using Machine-Learning," IoT, vol. 5, no. 4, pp. 634–649, 2024. Publisher: MDPI, Basel, Switzerland.
[4] M. D. S. Sharafat, N. D. Kabya, R. I. Emu, M. U. Ahmed, J. C. Onik, M. A. Islam, and R. Khan, "An IoT-enabled AI system for real-time crop prediction using soil and weather data in precision agriculture," Smart Agricultural Technology, vol. 12, Art. no. 101263, 2025. Publisher: Elsevier, Amsterdam, Netherlands.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 5, May 2026

|DOI: 10.15680/IJIRSET.2026.1505121|

[5] M. Rashid, B. S. Bari, Y. Yusup, M. A. Kamaruddin, and N. Khan, "A comprehensive review of crop yield prediction using machine learning approaches with special emphasis on palm oil yield prediction," IEEE Access, vol. 9, pp. 63406–63439, 2021. Publisher: IEEE, Piscataway, NJ, USA.
[6] S. Karetsos, G. Costopoulou, and M. Sideridis, "Developing a smartphone app for m-government in agriculture," Journal of Agricultural Informatics, vol. 5, no. 1, pp. 1–8, 2014.
[7] L. Ruiz-Garcia, L. Lunadei, P. Barreiro, and J. I. Robla, "A review of wireless sensor technologies and applications in agriculture and food industry: State of the art and current trends," Sensors, vol. 9, no. 6, pp. 4728–4750, 2009. Publisher: MDPI, Basel, Switzerland.
[8] N. Ahmed, D. De, and I. Hussain, "Internet of Things (IoT) for smart precision agriculture and farming in rural areas," IEEE Internet of Things Journal, vol. 5, no. 6, pp. 4890–4899, 2018. Publisher: IEEE, Piscataway, NJ, USA.
[9] P. P. Jayaraman, D. Palmer, A. Zaslavsky, and D. Georgakopoulos, "Do-it-yourself digital agriculture applications with semantically enhanced IoT platform," IEEE Intelligent Systems, vol. 30, no. 4, pp. 33–38, 2015. Publisher: IEEE, Piscataway, NJ, USA.
[10] A. Kamilaris, A. Kartakoullis, and F. X. Prenafeta-Boldú, "A review on the practice of big data analysis in agriculture," Computers and Electronics in Agriculture, vol. 143, pp. 23–37, 2017. Publisher: Elsevier, Amsterdam, Netherlands.
[11] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001. Publisher: Springer, Berlin, Germany.
[12] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, Aug. 2016, pp. 785–794.
[13] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, no. 3, pp. 273–297, 1995. Publisher: Springer, Berlin, Germany.
[14] D. Bzdok, M. Altman, and M. Krzywinski, "Statistics versus machine learning," Nature Methods, vol. 15, no. 4, pp. 233–234, 2018. Publisher: Nature Publishing Group, London, UK.
[15] J. Jones, M. Hoogenboom, and J. Antle, "The DSSAT cropping system model," European Journal of Agronomy, vol. 18, no. 3–4, pp. 235–265, 2003. Publisher: Elsevier, Amsterdam, Netherlands.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details