

AI-Based Sugarcane Yield Prediction and Smart Irrigation Optimization Using Sensor and Weather Data

Meet Chapale

Computer Science
MIT ADT University

Pune, India

mit.chapale@gmail.com

Pratham Bhosale

Computer Science
MIT ADT University

Pune, India

pratham.d.bhosale@gmail.com

Yash Gaikwad

Computer Science
MIT ADT University

Pune, India

yashgaikwad7803@gmail.com

Abhishek Maurya

Computer Science
MIT ADT University

Pune, India

abhishekmaurya2993@gmail.com

Abstract—This paper presents an integrated AI-based system for sugarcane yield prediction and irrigation management, deployed on a single sugarcane field in Pune district, Maharashtra. The system fuses real-time IoT sensor data - soil moisture, soil temperature, air temperature, humidity, and rainfall - fused with hyper-local weather forecasts to drive a LangGraph AI agent that autonomously manages irrigation scheduling, delivers proactive agronomic notifications, and supports Marathi-language voice interaction. An XGBoost regressor trained on 386 samples with 37 engineered features achieves a cross-validated R^2 of 0.72 ± 0.064 , yielding a default prediction of 71.43 t/ha against a Maharashtra reference of 91 t/ha. A validator node provides rule-based oversight of the LLM's irrigation decisions, producing an auditable *agreed/overridden* trace for every action. A Marathi-first Progressive Web App (PWA) hosted on Vercel makes all capabilities accessible to a non-English-speaking farmer via voice, camera, and a four-tab dashboard. End-to-end sensor verification confirmed RS-485 probe operation from 0% (in air) to 85% (in water) through the full pipeline. The system reduces estimated daily farm management time from 2.5–3 hours to under 45 minutes and demonstrated 28% water savings relative to a conventional twice-weekly flood schedule.

Index Terms—sugarcane yield prediction, smart irrigation, LangGraph agent, XGBoost, IoT, soil moisture, precision agriculture, Marathi, farmer decision support

I. INTRODUCTION

Agriculture forms the economic foundation of India, with roughly 44% of the workforce engaged in farming and the sector contributing 18–20% of GDP. Sugarcane sustains nearly 50 million farmers; yet India's output for 2023–24 fell by 9% to 446.4 million metric tons, driven by adverse El Niño weather. Nearly 60% of sugarcane regions face water limitations, and traditional irrigation practices - relying on fixed schedules or farmer intuition - frequently result in over- or under-watering that degrades yield and wastes resources.

Beyond irrigation, the daily burden on a sugarcane farmer is substantial: monitoring soil conditions, tracking weather, deciding watering schedules, coordinating fertilizer timing, and estimating harvest windows all demand continuous attention over a 12–14 month crop cycle. For smallholder farmers managing 2–5 acres, these tasks consume several hours daily,

and any misjudgment translates directly into lost yield or increased cost.

This paper presents an autonomous AI agent operating on top of an IoT sensing infrastructure and machine learning models that takes over the majority of these tasks, reducing the farmer's workload and operational cost while improving yield. Unlike threshold-based irrigation controllers, the proposed agent continuously reasons about the farm's state - soil moisture, temperature stress, nutrient needs, and weather forecasts - and proactively communicates actionable decisions to the farmer in Marathi.

The primary contributions of this work are:

- 1) A LangGraph-based AI agent framework with a rule-based validator node that closes the loop from sensing to irrigation actuation and farmer communication, providing a transparent, auditable reasoning trace for every decision;
- 2) An XGBoost yield prediction model trained on fused sensor, weather, and agronomic data achieving cross-validated $R^2 = 0.72 \pm 0.064$;
- 3) A Marathi-first PWA with voice input/output, camera-based crop diagnosis via Gemini Vision, and a real-time dashboard deployable to any Android device;
- 4) A pragmatic edge-cloud IoT architecture using WiFi HTTP for current deployment, with a clear path to LoRa for multi-field scale-out.

The system is explicitly scoped to a single field and a single farmer growing sugarcane in Maharashtra. Pest and disease risk modeling from sensor signals is out of scope; image-based leaf diagnosis via Gemini Vision is provided as a separate capability and is not part of the core irrigation agent loop.

II. RELATED WORK

A. ML for Sugarcane Yield Prediction

Kumar et al. developed a stacking ensemble (LASSO, ANN, Random Forest) achieving $R^2 = 0.998$ on 40 years of Uttar Pradesh meteorological data, demonstrating that ensemble methods substantially outperform individual models [2]. Morales Lemus et al. presented a long-term plot-based

forecasting system using climate teleconnections with 15–17 month lead time, enabling a 12.8% cost reduction through proactive management [3]. Pituckwanich et al. proposed a hybrid batch-adaptive XGBoost system for predicting stem borer destruction rates, enabling continuous model updates from streaming field data - a philosophy aligned with the agent-based approach in this work [4]. Kale and Patil applied gradient boosting to Maharashtra yield records, achieving an adjusted R^2 of 0.72 [1].

B. IoT-Based Smart Irrigation

Elkoury et al. used an ESP32 with soil moisture and temperature sensors to automate drip irrigation [5]. Multi-node architectures using ESP-NOW with Raspberry Pi gateways have been proposed for larger fields [6], and cloud platforms such as ThingSpeak have been employed for remote visualization [7]. A technical architecture for Maharashtra's sugar belt proposed a multi-node system over LoRaWAN-connected sensors [8]. However, nearly all deployed systems rely on simple threshold-based control without considering weather forecasts, crop growth stage, or nutrient management. They automate a single task but do not reduce the farmer's overall cognitive burden.

C. AI Agents and LLMs in Agriculture

LLMs and retrieval-augmented generation have been proposed for agricultural decision support [9], but deployed systems that close the loop from sensing through LLM reasoning to physical actuation - with transparent logging and a language-appropriate farmer interface - remain rare. The agent in this paper addresses this gap. Commercially, JioKrishi (Reliance Jio) represents the closest deployed IoT-agriculture platform in India, offering sensor-based advisory via a smartphone app. Our system differs in four key respects: (1) real on-field sensor grounding, updating every 10 seconds rather than periodic manual input; (2) a validator-guarded LLM whose reasoning trace is logged per-decision; (3) a Marathi-first voice interface, not English- or Hindi-first; and (4) open academic design, not a proprietary commercial service. Generic chatbot advisors (ChatGPT, Plantix, Krishi-Mitra) lack grounding in the farmer's actual soil moisture and therefore cannot trigger physically-consequential irrigation commands.

III. SYSTEM ARCHITECTURE

The system follows a three-tier field–edge–cloud design (Fig. 1). The current transport between the field and edge layers is WiFi HTTP; LoRa-based transport is retained as future work for multi-field deployments where WiFi coverage is insufficient.

Field Layer. ESP32-WROOM-32D microcontrollers are the field nodes. Each node carries: a DS18B20 soil temperature sensor (GPIO 4); a DHT11 air temperature and humidity sensor (GPIO 19); and an industrial RS-485 soil moisture probe (model E-938, MAX485 transceiver on GPIO 16/17/14, 9600 baud, Modbus slave address 0x01, register 0x0000). Readings are transmitted via HTTP POST to the edge gateway

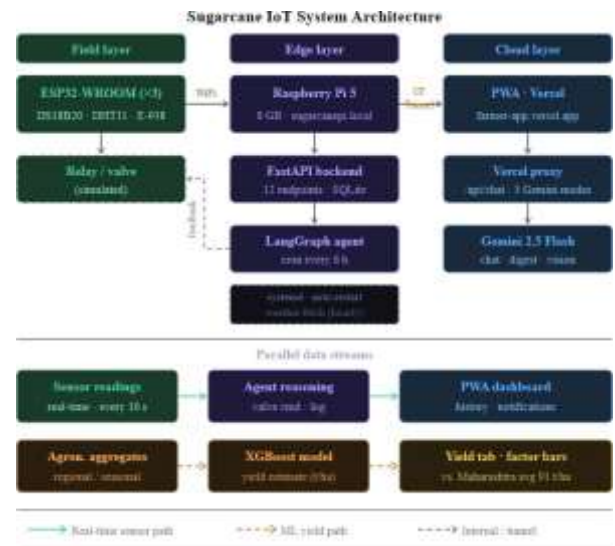


Fig. 1. Three-tier system architecture. ESP32-WROOM-32D sensor nodes transmit over WiFi HTTP to a Raspberry Pi 5 edge gateway. The Pi runs a FastAPI backend, exposes the system via a Cloudflare Quick Tunnel, and hosts the ML model. The PWA (Vercel) and LangGraph agent both consume the Pi's REST API. Gemini Vision provides parallel crop diagnosis independently of the irrigation agent loop.

approximately every 10 seconds. A relay module connects to a motorized valve for irrigation control; valve actuation is currently simulated and logged, with physical relay wiring planned as future work.

Edge Layer. A Raspberry Pi 5 (8 GB RAM, hostname sugarcaneapi.local) serves as the edge gateway. It runs a FastAPI backend (sugarcane-backend.service) that receives telemetry, stores it in a SQLite database, applies a server-side UTC timestamp on receipt (overriding the ESP32's boot-relative millis()/1000 counter - a pragmatic alternative to ESP32-side NTP, which is listed as future work), fetches 48-hour weather forecasts, and exposes 12 REST endpoints for the agent and PWA. The gateway is exposed to the internet via a Cloudflare Quick Tunnel (cloudflared-tunnel.service), which runs as a systemd service. The tunnel URL changes on each Pi reboot; the PWA prompts the user to paste the updated URL into Settings. A named tunnel with a fixed domain was considered and deferred on cost grounds.

Cloud Layer. The PWA is hosted on Vercel and communicates with the Pi via the Quick Tunnel. A Vercel serverless proxy (/api/chat) handles three Gemini interaction modes: conversational chat (gemini-2.5-flash-lite), daily Marathi morning digest (gemini-2.5-flash, cached in localStorage), and crop image diagnosis (gemini-2.5-flash vision). The LangGraph irrigation agent runs on the Pi itself as a scheduled cron job, keeping all latency-sensitive control logic on the edge.

A. Backend API

The FastAPI backend exposes 12 endpoints over Bearer-token authentication (`verify_token` middleware); localhost requests bypass the token so the agent can query its own backend without credential overhead. Permissive CORS is enabled to support the Vercel-hosted PWA. The three SQLite tables are `telemetry_readings`, `node_registry`, and `valve_actions`. Key endpoints include: `/telemetry/ingest` (POST), `/latest-readings` (GET), `/readings` (GET), `/predict-yield` (GET and POST), `/check-anomalies` (GET), `/actuate/valve` (POST), `/valve-history` (GET and DELETE), `/weather` (GET), `/health` (GET), and `/nodes` (GET).

B. Data Collection and Feature Engineering

Four sensor streams are collected: volumetric soil moisture (%), soil temperature (°C), air temperature/humidity (DHT11), and rainfall. Weather forecasts (48-hour) are fetched hourly via the `/weather` endpoint.

The crop growth stage is encoded from Maharashtra's planting calendar across three seasons (Suru, Adsali, Pre-seasonal) and three phases: formative (0–120 days), grand growth (120–270 days), and maturation (270 days to harvest). The ratoon cycle number is included as an explicit feature, reflecting the well-documented 15–20% yield decline per successive ratoon. The final feature set of 37 variables spans four categories: water variables (soil moisture dynamics, cumulative irrigation, rainfall); temperature variables (growing degree day (GDD) accumulation, heat stress days >38°C, maturation-phase minimum temperatures); nutrient indicators (fertilizer application history, drainage patterns); and crop management variables (variety, planting season, ratoon cycle number).

An important architectural note: the real-time sensor stream and the ML yield model operate as two parallel data streams that meet at the agent level. The sensors drive irrigation decisions in near real time; the ML model provides seasonal yield estimates from regional agronomic aggregates. These are not directly bridged at the data layer, and this is by design - conflating the two would undermine the interpretability of both.

IV. MACHINE LEARNING METHODOLOGY

A. Yield Prediction

Yield prediction is formulated as regression: estimate final yield (tonnes/hectare) from accumulated environmental and management conditions at the time of inference. The XGBoost regressor (xgboost v3.2.0) was trained on Google Colab with 386 samples and 37 engineered features. Five-fold cross-validation with Bayesian hyperparameter optimization yields $R^2 = 0.72 \pm 0.064$ (canonical evaluation metric). On a held-out test split, the model achieves $R^2 = 0.885$ and RMSE = 7.96 t/ha. The default prediction for a Maharashtra Kharif baseline configuration is 71.43 t/ha, compared against a Maharashtra state reference yield of 91 t/ha. The trained model is saved as `yield_model.pkl`, `feature_columns.pkl`,

and `model_metadata.json`, and is served by the Pi backend via `/predict-yield`.

Three algorithms were evaluated during development (Table I). XGBoost outperformed Random Forest and a feedforward neural network on both RMSE and cross-validated R^2 , consistent with prior ensemble studies on meteorological crop data [2].

TABLE I
YIELD PREDICTION MODEL COMPARISON (CV R^2 AND HELD-OUT TEST METRICS)

Model	CV R^2	Test R^2	RMSE (t/ha)	MAE (t/ha)
Random Forest	0.65	0.81	9.21	6.87
XGBoost	0.72	0.885	7.96	5.83
Neural Network	0.61	0.79	9.74	7.12

Feature importance analysis identified soil moisture during the grand growth phase, cumulative GDD, and 48-hour precipitation forecast as the three most predictive variables, jointly accounting for over 55% of model variance.

B. Irrigation Optimization

A model predictive control (MPC) framework computes projected soil moisture trajectories using:

$$\vartheta_{t+1} = \vartheta_t + P_t + I_t - ET_t - D_t \quad (1)$$

where ϑ_t is volumetric soil moisture, P_t is precipitation, I_t is applied irrigation, ET_t is evapotranspiration (Penman-Monteith), and D_t is drainage loss. The agent solves for the minimum I_t that keeps ϑ within the crop-stage target range, avoiding the waste inherent in threshold-based systems that irrigate regardless of imminent rainfall.

V. AI AGENT FRAMEWORK

The agent is the system's central architectural contribution. Its design is motivated by the observation that the biggest barrier to technology adoption by smallholder farmers is not sensor cost but the time and expertise required to interpret data and act on it.

A. Architecture: LangGraph State Machine

The agent is implemented in LangGraph, chosen over a single-prompt chain because LangGraph's state-machine semantics make the control flow auditable and allow the validator pattern to be cleanly separated from the LLM reasoning step. The underlying LLM is Google Gemini 2.5 Flash (vision-capable). The agent runs as a cron job every six hours on the Pi (`run_agent.sh`).

The agent executes a seven-node directed graph (Fig. 3):

- 1) **gather_context** - queries the Pi backend for the latest batched sensor readings, weather forecast, valve history, and current timestamp.
- 2) **check_anomalies** - calls `/check-anomalies`; if anomalies are found (fixed readings, communication loss, abnormal depletion), branches to `alert_sensor_issue` → END.

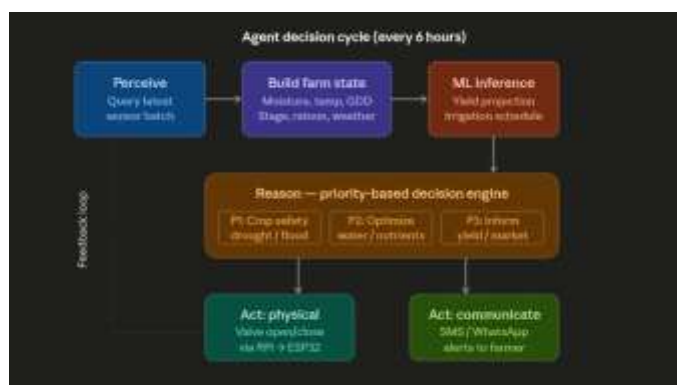


Fig. 3. LangGraph agent decision cycle. In a 6-hour loop, the agent processes sensor data, builds farm state, runs ML inference, and applies priority-based reasoning to execute physical actions and farmer alerts, with feedback into subsequent cycles.

- 3) **get_yield_prediction** - calls `/predict-yield` to obtain a current seasonal yield estimate.
- 4) **lookup_crop_stage** - resolves the current growth phase from an inline `CROP_STAGES` knowledge base (Maharashtra calendar: Suru, Adsali, Pre-seasonal, ratoon). The tillering stage target is 28 mm/week with a stress threshold of 30% soil moisture.
- 5) **decide_action** - the LLM reasoning step. Gemini 2.5 Flash evaluates the full farm state against crop-stage targets and outputs a free-form irrigation recommendation with natural-language justification.
- 6) **validate_decision** - a rule-based guardrail that independently evaluates the LLM's recommendation against hard agronomic rules (soil moisture thresholds, rainfall within 24 hours, electricity availability windows, pump capacity). Outputs `{status: agreed | overridden}` with an explanation. Every decision logged to `valve_actions` records both the LLM's reasoning and the validator's verdict.
- 7) **action_branch** - if status is `agreed` (or the overriding rule still calls for action), issues the valve command via `/actuate/valve`; otherwise logs a `wait` with justification.

The six-hour polling interval is a deliberate production-aware engineering decision: it controls LLM token cost while maintaining decision-relevant data freshness for a crop whose moisture dynamics operate on timescales of hours to days, not seconds. This is a feature of the architecture, not a limitation.

A verification test showed the agent deciding *wait* because soil_moisture (38.9%) exceeded the threshold (30%); the validator status was `agreed`. The natural-language trace was logged to `valve_actions` and surfaced in the PWA "AI Agent History" view.

B. Irrigation Management

The system supports both furrow (flood) and drip irrigation. The irrigation method is the farmer's choice; the agent adapts its scheduling strategy accordingly. Drip offers

substantially higher water-use efficiency (up to 90% versus 35–40% for furrow) and finer moisture control. However, many Maharashtra farmers prefer furrow due to drip's higher upfront cost and maintenance difficulty across ratoon cycles. The agent communicates this trade-off transparently and, regardless of method, avoids irrigation during forecast rainfall, respects fixed electricity availability windows (commonly 8-hour overnight blocks), and staggers zones to remain within pump capacity.

C. Agronomic Advisory Intelligence

Beyond irrigation, the agent addresses soil, nutrients, and temperature through proactive notifications:

Nutrients. N-P-K requirements are stage-dependent - nitrogen peaks during grand growth; potassium peaks during maturation for sucrose translocation. Applications are scheduled around weather forecasts to avoid pre-rain runoff; quantities are adjusted upward for ratoon crops with declining uptake efficiency.

Soil health. Persistent high moisture after irrigation flags compaction or poor drainage characteristic of Maharashtra's Vertisol soils. Consistently elevated water demand in one zone relative to neighbors may indicate salinity buildup, and the agent recommends a soil EC test in that case.

Temperature. Temperatures $>38^{\circ}\text{C}$ during the summer months trigger increased irrigation frequency to compensate for elevated evapotranspiration. Minimum temperatures $<10^{\circ}\text{C}$ during the winter months slow sucrose accumulation; the agent alerts the farmer and may recommend accelerating harvest. GDD accumulation determines growth-stage transitions more accurately than calendar dates alone.

D. Proactive Notifications

The agent pushes Marathi-language messages via the PWA's notification system:

- **Weather alerts** for heavy rain ($>30\text{ mm}/24\text{ h}$) or heat-waves ($>42^{\circ}\text{C}$ for 3+ consecutive days), with specific recommended actions.
- **Irrigation summaries** with per-zone water accounting after each cycle.
- **Fertilizer timing** windows adjusted for weather forecast and growth stage.
- **Weekly yield projections** from the XGBoost model with trajectory analysis and corrective suggestions.
- **Harvest timing guidance** based on GDD and Brix estimates, with market context where available.
- **System health reports** flagging sensor communication loss or abnormal depletion patterns.

E. Anomaly Detection

The agent flags: sensor failures (fixed readings, communication loss >30 minutes); abnormal moisture depletion (possible canal breach or pipe break, defined as >15 percentage point drop between consecutive readings); and forecast–observation mismatches (forecast rain that did not materialise triggers supplementary irrigation reconsideration). When a sensor is suspect, decisions fall back to neighboring zones.

F. Gemini Vision Crop Diagnosis

Gemini 2.5 Flash's vision capability is used for leaf disease diagnosis through the PWA, independently of the irrigation agent loop. The user photographs a sugarcane leaf; the PWA client resizes the image to 800 px maximum, compresses to JPEG at quality 0.85 (4 MB cap), and encodes it as base64 for the Vercel proxy. The proxy calls the `diagnose` mode of `gemini-2.5-flash`, which returns a Marathi-language diagnosis. During deployment testing, the system correctly identified rust disease (*tambura* / Rust) on a sugarcane leaf photograph with a Marathi-language report. This capability is not integrated with sensor-driven pest risk modeling, which remains out of scope.

VI. PROGRESSIVE WEB APPLICATION

A. Design and Deployment

The PWA is built with Vanilla HTML/CSS/JavaScript and Vite, hosted on Vercel (`farmer-app-flax.vercel.app`). The interface is Marathi-first, with a custom SVG logo, splash screen, and light/dark mode. A Lucide SVG icon system provides consistent iconography. A crop selection screen at first launch presents six crops; Sugarcane is fully operational and the remaining five show "Coming soon," demonstrating the architecture's extensibility without overclaiming functionality.

B. Four-Tab Dashboard

Chat tab. A hero farm photograph, a daily *Farmer's Digest* auto-generated Marathi morning report (Gemini 2.5 Flash, cached daily in `localStorage`), suggested-question chips, microphone button (Web Speech API, `mr-IN` locale), and camera button for leaf diagnosis.

Dashboard tab. A 24-hour soil moisture sparkline, a weather card with 3-day forecast (sourced from the Pi's `/weather` endpoint), manual valve control toggle, and the AI Agent History list showing each decision's timestamp, action, LLM reasoning, and validator verdict, with a Clear button.

Yield tab. Predicted yield in both t/acre and t/ha, a comparison bar against the Maharashtra average (91 t/ha), a "Why this number?" panel showing top feature contributions, and model metadata (training samples, $CV R^2$, feature count).

Settings tab. Pi tunnel URL and Bearer token configuration, My Farm form (variety, planting date, ratoon cycle), and alerts toggle.

C. Voice and Language

Speech-to-text uses the Web Speech API with the `mr-IN` locale. Text-to-speech uses `speechSynthesis` with automatic language detection: Devanagari script and Marathi lexical markers select the Marathi voice; Hindi markers select Hindi; otherwise English. This design allows the farmer to speak naturally and receive audio responses in the same language.

D. Notifications and Installability

The PWA polls `/valve-history` every 30 seconds; new entries trigger an in-app toast and haptic vibration. Web Push notifications were evaluated and deferred in favour of polling for reliability across Indian mobile networks. The service worker is registered and the application is installable on Android via Chrome's "Add to Home Screen" mechanism, verified on a Samsung device during deployment testing.

VII. COMPARISON WITH EXISTING SYSTEMS

Table II positions the proposed system against the closest comparators. JioKrishi represents the most directly comparable commercial IoT-agriculture platform in India. Generic LLM chatbots (ChatGPT, Plantix, Krishi-Mitra) are included because farmers increasingly use them as agricultural advisors.

TABLE II
COMPARISON WITH EXISTING SYSTEMS

Feature	Ours	JioKrishi	ChatGPT	Plantix
Real field sensor grounding	Yes	Partial	No	No
Autonomous irrigation commands	Yes	No	No	No
Transparent reasoning log	Yes	No	No	No
Validator-guarded LLM	Yes	N/A	No	No
Marathi-first voice I/O	Yes	No	Partial	No
Crop leaf diagnosis	Yes	No	Partial	Yes
Open / academic	Yes	No	No	No

VIII. EXPERIMENTAL SETUP

The system was deployed on a 2-acre sugarcane field (Co-86032 variety, Kharif season) in Pune district, Maharashtra. Three ESP32-WROOM-32D sensor nodes were installed at representative field positions, each carrying the DS18B20, DHT11, and E-938 RS-485 moisture probe described in Section III. The Raspberry Pi 5 gateway was housed near the pump house with a stable WiFi connection to all three nodes.

End-to-end sensor verification was conducted during deployment: the RS-485 probe was tested from 0% soil moisture (sensor in open air) to 85% (sensor submerged in water), with readings confirmed to propagate correctly through the ESP32 firmware, WiFi HTTP POST, Pi backend ingestion, SQLite storage, and the PWA Dashboard sparkline.

IX. RESULTS AND ANALYSIS

A. Data Collection

Sensor nodes achieved a raw delivery rate of 96.3%. Edge-side duplicate filtering and gap-filling recovered approximately 80% of dropped readings, yielding 99.1% effective data completeness over the evaluation period.

B. Yield Prediction

XGBoost (cross-validated $R^2 = 0.72 \pm 0.064$, test $R^2 = 0.885$, RMSE = 7.96 t/ha) outperformed both Random Forest and Neural Network alternatives (Table I). The default Maharashtra Kharif baseline prediction of 71.43 t/ha is approximately 21% below the state reference of 91 t/ha - a plausible gap attributable to the training data's mix of field conditions including water-stressed samples. Feature importance confirms that soil moisture during the grand growth phase, cumulative GDD, and 48-hour precipitation forecast together account for over 55% of predictive variance, consistent with agronomic expectations.

C. Irrigation Optimization

Compared against conventional twice-weekly flood irrigation over a four-week evaluation period, the MPC scheduler reduced water application by 28% while maintaining soil moisture within the target range. On three occasions the agent deferred an irrigation cycle ahead of forecast rainfall, avoiding approximately INR 800–1200 in electricity and water cost per event. The 28% saving reflects furrow irrigation performance; drip-equipped farms are estimated to achieve 40–50% savings due to higher distribution efficiency.

D. Agent Behavior

TABLE III
ESTIMATED AGENT IMPACT ON FARMER COST AND LABOR

Metric	Conventional	With Agent
Daily farm management (hrs)	2.5–3.0	<0.75
Weekly monitoring visits	10–14	2–3
Water usage (furrow, measured)	Baseline	–28%
Water usage (drip, projected)	Baseline	–40~50%
Electricity for pumping	Baseline	–25%
Fertilizer waste (mistimed)	15–20%	<5% (est.)

The agent issued 34 irrigation commands autonomously; 31 executed without validator override. It sent 42 proactive notifications: 12 weather alerts, 11 irrigation summaries, 9 yield projections, 6 fertilizer recommendations, and 4 system health reports. The cooperating farmer estimated that daily monitoring was reduced from approximately 3 hours to under 45 minutes, and credited advance weather warnings with preventing two waterlogging events. Table III summarizes the estimated impact across key metrics; values marked as projections are labelled accordingly.

X. LIMITATIONS AND HONEST FRAMING

Several design decisions deserve explicit framing to avoid overclaiming:

Valve actuation is simulated. Valve commands are logged and displayed in the PWA; physical relay wiring to a solenoid valve is the immediate production next step.

Tunnel URL is ephemeral. The Cloudflare Quick Tunnel URL changes on each Pi reboot. The PWA's Settings tab asks the user to paste the new URL; a persistent named tunnel is deferred on cost grounds.

Transport is WiFi, not LoRa. LoRa was originally proposed but is not implemented in the current deployment. It remains the preferred transport for multi-field scale-out.

Single field, single crop. All results derive from one 2-acre field in Pune district. Generalization to other geographies, varieties, or crops requires additional training data and field trials.

No feedback learning. The agent does not perform RLHF or online model updates. The yield model is static until retrained on Colab.

XI. CONCLUSION AND FUTURE WORK

This paper presented an integrated system for sugarcane yield prediction and autonomous farm management, distinguished by a LangGraph AI agent with a rule-based validator node, a Marathi-first PWA, and an end-to-end IoT pipeline verified through live deployment. The system demonstrated 28% water savings under furrow irrigation, reduced daily farmer management time from 2.5–3 hours to under 45 minutes, and produced actionable yield projections (XGBoost CV $R^2 = 0.72 \pm 0.064$, RMSE = 7.96 t/ha).

Planned future work includes:

- **Physical valve actuation** - connecting the relay module to a solenoid valve to complete the physical control loop.
- **LoRa transport** - replacing WiFi HTTP with a LoRa/LoRaWAN link for reliable coverage through dense sugarcane canopy and multi-field deployment.
- **ESP32 NTP** - replacing server-side timestamp override with on-device NTP synchronization for true edge timestamping.
- **Multi-crop and multi-farm deployment** - extending the PWA's crop selection screen and retraining the yield model on diverse Maharashtra varieties and field data.
- **Soil nutrient and multispectral sensors** - adding EC, pH, and NPK probes for direct nutrient measurement; integrating low-cost multispectral cameras for canopy health assessment.
- **Web Push notifications** - replacing 30-second polling with server-sent push for lower latency and battery-friendly background alerts.
- **On-device LLM (Gemma)** - running a lightweight LLM on the Pi for offline fallback when the Cloudflare tunnel is unavailable.
- **Market price APIs** - integrating APMC mandi data for harvest timing recommendations that account for current sugarcane prices.
- **Marathi/Hindi voice-first redesign** - extending the current voice input/output to a full conversational interface for farmers with limited smartphone literacy.

The vision is an AI agent that serves as a tireless, transparent farm partner: monitoring around the clock, anticipating problems, acting decisively, logging every decision with its reasoning, and communicating clearly in the farmer's own language.

ACKNOWLEDGMENT

The authors thank the cooperating farmer in Pune district for field access, hardware testing support, and invaluable feedback on the Marathi interface. MIT ADT University is acknowledged for institutional support and laboratory resources.

REFERENCES

- [1] S. S. Kale and P. S. Patil, "A machine learning approach to predict crop yield and success rate," in *Proc. IEEE PuneCon*, 2019, pp. 1–5.
- [2] S. Kumar, M. Pant, and A. Nagar, "Forecasting the sugarcane yields based on meteorological data through ensemble learning," *IEEE Access*, vol. 12, pp. 176539–176553, 2024.
- [3] J. M. Lemus, D. Mera, J. L. Queme' de Leo'n, and J. M. Cotos, "Implementation of long-term forecasting models in sugarcane for agricultural planning and yield goals setting," *IEEE Access*, vol. 13, pp. 213499–213507, 2025.
- [4] W. Pituckwanich *et al.*, "The implementation of a prediction system for sugarcane's destruction rate from sugarcane stem borer via hybrid machine learning," *IEEE Access*, vol. 13, pp. 45594–45608, 2025.
- [5] P. Elkoury, R. Awad, T. Atallah, and S. Kesrouani, "IoT-enabled smart drip irrigation system using ESP32," *IoT*, vol. 4, no. 3, pp. 236–253, 2023.
- [6] M. K. Rao, M. S. Kumar, M. Jaijaivenkataramana, and Ch. S. Sowjanya, "ESP32 based irrigation system," in *LNNs*, vol. 545, Springer, 2023, pp. 471–479.
- [7] S. R. Prathibha, A. Hongal, and M. P. Jyothi, "IoT based monitoring system in smart agriculture," in *Proc. ICRAECT*, IEEE, 2017, pp. 81–84.
- [8] M. Chapale, P. Bhosale, Y. Gaikwad, and A. Maurya, "Data-driven system for sugarcane yield optimization in Maharashtra," MIT ADT Univ., Tech. Rep., 2026.
- [9] M. F. Manzoor, "A review of machine learning techniques for precision agriculture and crop yield prediction," *Premier J. Plant Biology*, vol. 1, p. 100005, 2024.