



AI-Enabled Sensor-Based Crop Stress Early Prediction and Detection System

Kanishkumar R¹, Sivasankar D², Dr. E. Sivanantham³

Student, Department of Artificial Intelligence and Data Science, Dhanalakshmi Srinivasan College of Engineering & Technology, Chennai, India^{1,2}

Professor, Department of Artificial Intelligence and Data Science, Dhanalakshmi Srinivasan College of Engineering & Technology, Chennai, India³

Abstract: Timely identification of crop stress is critical to reducing agricultural losses and supporting sustainable food production. This paper presents an AI-enabled sensor-based platform that integrates Internet of Things (IoT) sensing, cloud-based data management, and deep learning inference to predict and detect crop stress at an early stage. Temperature, humidity, and soil moisture data are continuously acquired using a DHT11 sensor and a capacitive soil moisture sensor interfaced with an ESP32 microcontroller, and the readings are transmitted to a Firebase Realtime Database. A Flask-based backend retrieves the stored time-series records and feeds them to a trained Long Short-Term Memory (LSTM) network, which forecasts future soil moisture values. The system concurrently analyses moisture trend behaviour—classifying conditions as stable, gradually decreasing, or critically decreasing—and cross-references actual and predicted values against crop-specific thresholds to generate three-level stress alerts: Low, Medium, and High. Early-warning notifications are raised when predicted values approach critical limits, while High-stress alerts are triggered when readings fall below safety thresholds. An interactive web dashboard delivers live sensor telemetry, LSTM forecast outputs, trend indicators, and stress alert status in real time. Experimental evaluation confirms accurate prediction and reliable alert generation, demonstrating that the proposed system provides an affordable, scalable, and intelligent solution for modern precision agriculture.

Keywords: crop stress detection; LSTM time-series forecasting; IoT smart agriculture; soil moisture prediction; ESP32; Firebase; precision farming; early warning system

I. INTRODUCTION

Crop stress caused by insufficient or excessive soil moisture, thermal extremes, or humidity imbalances is one of the leading causes of yield loss worldwide. Conventional field monitoring relies on periodic manual inspection, making it difficult to detect deteriorating conditions before visible damage appears. The combination of low-cost IoT sensors, cloud computing, and machine learning now offers a practical path toward continuous, autonomous crop health surveillance.

This paper proposes the AI-Enabled Sensor-Based Crop Stress Early Prediction and Detection System—a unified platform that collects environmental data through an ESP32 microcontroller fitted with DHT11 and soil moisture sensors, stores readings in a Firebase Realtime Database, and performs predictive analysis using an LSTM neural network deployed on a Flask backend. The system classifies stress severity into three levels and delivers alerts through a browser-based dashboard, enabling farmers to take corrective action before irreversible crop damage occurs.

The principal contributions of this work are: (i) a lightweight, low-cost IoT data acquisition layer built around the ESP32; (ii) a cloud-native data pipeline connecting edge hardware to a web interface; (iii) an LSTM model for multi-step soil moisture forecasting; (iv) a trend-aware alert engine that combines predicted values with crop-specific thresholds; and (v) a real-time monitoring dashboard with three-tier stress classification. The system aligns with the United Nations Sustainable Development Goals SDG 2 (Zero Hunger) and SDG 9 (Industry, Innovation, and Infrastructure).

II. RELATED WORK

Yasmeen and Ahmed [1] surveyed machine learning algorithms—Decision Trees, Random Forest, SVM, and shallow neural networks—for early crop stress classification using soil and climate datasets. Although classification accuracy was strong, the study did not integrate IoT hardware, cloud connectivity, or sequential forecasting, limiting applicability to offline scenarios.

Sharafat et al. [2] combined IoT environmental sensors with AI-based crop prediction in a precision agriculture setting. The architecture demonstrated reliable cloud data transmission and automated decision support, but omitted LSTM-based time-series forecasting and multi-level stress alert generation.

Suresh Kumar et al. [3] proposed a remote-sensing framework that uses multi-source satellite and UAV imagery with ML classifiers for plant stress identification. High spatial coverage was achieved; however, the approach depends on



expensive remote sensing infrastructure and lacks direct ground-level IoT integration or real-time alerting.

Patel et al. [4] deployed soil moisture, temperature, and humidity sensors with a cloud monitoring backend and applied ML-based irrigation classification. The work confirmed the benefit of cloud-enabled environmental data analysis but stopped short of deep learning forecasting or threshold-adaptive alert mechanisms.

Singh et al. [5] demonstrated deep learning for crop stress prediction using sequential sensor data and showed performance gains over traditional ML baselines. The proposed architecture, however, did not include a centralised cloud database, a user-facing dashboard, or a structured alert management module.

Table I consolidates the feature comparison across the surveyed systems and highlights the gaps addressed by the proposed work.

Table I: Comparison of Existing Smart Agriculture Systems

| System | IoT Sensors | LSTM Prediction | Cloud DB | Real-Time Alerts | Dashboard |
|-------------------------|-------------|-----------------|----------|------------------|-----------|
| Yasmeen et al. [1] | No | No | No | No | No |
| Sharafat et al. [2] | Yes | No | Yes | No | Partial |
| Suresh Kumar et al. [3] | Yes | No | No | No | No |
| Patel et al. [4] | Yes | No | Yes | No | No |
| Singh et al. [5] | Yes | No | No | Yes | Partial |
| Proposed System | Yes | Yes | Yes | Yes | Yes |

III. SYSTEM ARCHITECTURE

The proposed system follows a four-layer architecture: (i) Data Acquisition Layer, (ii) Cloud Storage Layer, (iii) AI Inference Layer, and (iv) Presentation Layer.

A. Data Acquisition Layer

An ESP32 microcontroller serves as the edge device. A DHT11 sensor measures ambient temperature and relative humidity, while a capacitive soil moisture sensor provides volumetric moisture percentage. Readings are sampled every 20 seconds. A 16×2 I²C LCD module displays live values locally. After WiFi association, the ESP32 pushes structured JSON payloads to the Firebase Realtime Database at /sensor/temperature, /sensor/humidity, and /sensor/soil_moisture.

B. Cloud Storage Layer

Firebase Realtime Database provides a serverless, low-latency key-value store that enables simultaneous read access by the Flask backend and the dashboard. A time-stamped history log is maintained under /sensor/history to support LSTM training and inference. Firebase's built-in security rules restrict write access to authenticated ESP32 clients and read access to the backend service.

C. AI Inference Layer

A Python Flask server retrieves the most recent 50 time-stamped sensor records from Firebase and reshapes them into (sequence_length × features) tensors. The LSTM model, trained in PyTorch on historical field data, accepts sequences of [temperature, humidity, soil_moisture] triplets and forecasts the next-step soil moisture value. In parallel, a rule-based trend analyser computes the slope of the last ten moisture readings and classifies behaviour as Stable ($|\text{slope}| < 0.5$), Gradually Decreasing ($-2.0 \leq \text{slope} < -0.5$), or Critically Decreasing ($\text{slope} < -2.0$). The stress level is determined by cross-referencing predicted moisture, actual moisture, and crop-specific minimum thresholds.

D. Presentation Layer

A Jinja2-templated HTML dashboard served by Flask displays current sensor readings, the LSTM-predicted next moisture value, trend status, and stress alert level. JavaScript polling refreshes data every 10 seconds without requiring page reloads, giving users a near-real-time view of field conditions.

IV. METHODOLOGY

A. Dataset and Preprocessing

Training data consist of 72-hour continuous sensor logs captured from a controlled test bed containing soil of uniform type and a single potted crop. Each record contains a UNIX timestamp, temperature (°C), humidity (%), and soil moisture



(%). Records with sensor read errors (NaN) are dropped and missing intervals are linearly interpolated. Values are normalised to [0, 1] using per-feature min-max scaling before being fed to the LSTM.

B. LSTM Model Design

The LSTM model contains one recurrent layer with 64 hidden units followed by a fully connected linear layer that produces a single moisture forecast. Input sequences of length 10 are constructed using a sliding window with stride 1. The model is trained for 100 epochs using the Adam optimiser ($\text{lr} = 0.001$) and MSE loss. Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and the coefficient of determination (R^2) are used as evaluation metrics.

C. Stress Alert Engine

Let M_{actual} denote the current soil moisture reading, M_{pred} the LSTM one-step forecast, and T_{min} the crop-specific minimum safe moisture threshold. Three alert levels are defined as follows: (i) Low Stress — $M_{\text{actual}} > T_{\text{min}}$ and $M_{\text{pred}} > T_{\text{min}}$ and trend is Stable; (ii) Medium Stress — M_{pred} approaches T_{min} (within 10%) or trend is Gradually Decreasing; (iii) High Stress — $M_{\text{actual}} \leq T_{\text{min}}$ or $M_{\text{pred}} \leq T_{\text{min}}$, or trend is Critically Decreasing. This hierarchical logic ensures that farmers receive actionable warnings before the crop enters a critical state.

V. IMPLEMENTATION

A. Hardware Setup

The prototype uses an ESP32 DevKit V1, a DHT11 digital temperature–humidity sensor connected to GPIO 4, a resistive soil moisture sensor mapped to ADC pin GPIO 34, and a 16×2 I²C LCD at address 0x27. The hardware components are housed on a breadboard for experimental convenience and powered through a USB power bank to simulate field deployment.

B. Software Stack

The complete software stack is summarised below:

Edge firmware: Arduino C++ with `Firebase_ESP_Client` and `DHT` libraries

Cloud backend: Google Firebase Realtime Database (NoSQL JSON store)

AI inference server: Python 3.10, Flask 2.3, PyTorch 2.0

LSTM training: PyTorch `nn.LSTM` with Adam optimiser

Frontend dashboard: HTML5, CSS3, JavaScript (Fetch API polling)

Deployment: Local Flask server; portable to any cloud VM

C. Key Code Modules

The ESP32 firmware reads DHT11 and ADC soil moisture every 20 seconds, maps raw ADC values (0–4095) to moisture percentage (0–100%) using `Arduino map()`, and pushes the three float/int values to Firebase. The Flask backend exposes a `/dashboard` route that retrieves the last 50 records, reconstructs the PyTorch LSTM model from a saved checkpoint (`crop_stress_model.pth`), runs inference, evaluates the trend rule engine, and renders the Jinja2 template with context variables `sensor`, `prediction`, `trend`, and `alert`.

VI. RESULTS AND DISCUSSION

The LSTM model was evaluated on a held-out 20% test split. The model achieved a MAE of 1.83%, RMSE of 2.41%, and R^2 of 0.94, indicating high predictive accuracy for single-step soil moisture forecasting. Training loss converged within 60 epochs, demonstrating stable optimisation behaviour.

Alert accuracy was assessed over a 24-hour live monitoring session with intentional moisture depletion cycles introduced by withholding irrigation at known intervals. The system correctly issued Medium Stress alerts 8–12 minutes before soil moisture crossed the Low Stress boundary in all five depletion cycles, and correctly issued High Stress alerts within one inference cycle (20 seconds) of the moisture falling below the threshold. No false High Stress alerts were recorded.

Dashboard refresh latency averaged 1.2 seconds end-to-end (ESP32 → Firebase → Flask → Browser), well within the 10-second polling interval. Firebase connection reliability was 99.7% over the 24-hour session, with two brief disconnections recovered automatically by the ESP32 reconnection handler.

Compared to the surveyed systems (Table I), the proposed platform is the only architecture that simultaneously integrates real-time IoT sensing, cloud-native storage, LSTM forecasting, trend analysis, multi-level alerting, and dashboard visualisation in a single deployable stack—at a hardware cost below USD 15.

VII. CONCLUSION AND FUTURE WORK

This paper presented an AI-enabled, sensor-based crop stress early prediction and detection system that combines IoT edge hardware, Firebase cloud storage, LSTM time-series forecasting, and a trend-aware alert engine into a cohesive



precision agriculture platform. Experimental results confirmed high prediction accuracy ($R^2 = 0.94$) and timely, reliable stress alerting. The system offers an affordable and technically accessible solution for small-scale farmers and agricultural researchers.

Future work will explore multi-sensor fusion incorporating NPK soil probes and leaf wetness sensors to extend stress classification beyond moisture-centric conditions. Transfer learning from larger open agricultural datasets will be investigated to improve generalisation across soil types and crop varieties. Solar-powered, battery-backed hardware packaging and a companion mobile application are planned for field-scale pilot deployment. Integration with automated micro-irrigation actuators will close the feedback loop from detection to intervention.

ACKNOWLEDGEMENT

The authors thank the Department of Artificial Intelligence and Data Science, Dhanalakshmi Srinivasan College of Engineering & Technology, Chennai, for providing laboratory infrastructure and technical guidance. They also acknowledge the support of project guide Dr. E. Sivanantham and Head of Department Dr. A. Samuel Chellathurai throughout this work.

REFERENCES

- [1] G. Yasmeen and T. Ahmed, "A comprehensive study on machine learning-based techniques for early crop stress detection," *Int. J. Smart Agriculture AI Syst.*, 2025.
- [2] M. D. S. Sharafat, N. D. Kabya, R. I. Emu et al., "An IoT-enabled AI system for real-time crop prediction using soil and weather data in precision agriculture," *J. Precision Agric. Technol.*, 2025.
- [3] L. K. Suresh Kumar, R. Uyyala, A. R. Lakshmi et al., "AI-driven framework for early detection of plant stress using multi-source remote sensing data and machine learning," in *Proc. Int. Conf. Smart Farming AI Appl.*, 2025.
- [4] N. Patel, R. Shah, and V. Patel, "Smart agriculture monitoring using IoT and machine learning techniques," *Int. J. Agric. Technol.*, vol. 11, no. 3, pp. 210–225, 2022.
- [5] A. Singh, P. Kumar, and R. Sharma, "Crop stress prediction using deep learning and IoT-based monitoring systems," in *Proc. Int. Conf. Smart Agric. Technol.*, 2023.
- [6] Google LLC, *Firestore Realtime Database Documentation*, Firebase Platform, 2025. [Online]. Available: <https://firebase.google.com/docs/database>
- [7] PyTorch Foundation, *PyTorch Documentation*, 2025. [Online]. Available: <https://pytorch.org/docs>
- [8] Espressif Systems, *ESP32 Series Datasheet and IoT Development Documentation*, 2025. [Online]. Available: <https://www.espressif.com/en/products/socs/esp32>
- [9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
- [10] United Nations, "The 17 Goals: Sustainable Development Goals," UN Department of Economic and Social Affairs, 2015. [Online]. Available: <https://sdgs.un.org/goals>