AI-Based Decision Support System for Precision Irrigation Using IoT Sensor Data

[1]Sonal Sharma, [2]Shashi Kant Gupta

[1]Lincoln University College, Malaysia
[2]Adjunct Professor, Lincoln University College, Malaysia

Corresponding author: pdf.sonal@lincoln.edu.my, s.sonal@jainuniversity.ac.in

Abstract: Precision farming has a timing problem—water at the wrong moment and you either stress the crop or waste a resource you can't get back, and more and more, the job of getting that timing right is falling to automated decision systems rather than guesswork. IoT sensors are part of the answer, streaming soil and climate readings around the clock, but a stream of numbers doesn't irrigate anything on its own; it needs a layer on top that reads the conditions and decides. That layer is what this paper builds: an AI-based decision support system for precision irrigation that runs on live IoT sensor data. The model is deliberately lightweight—trained on past readings of soil moisture, soil temperature, and relative humidity, it learns to forecast when a crop needs water, and we pose irrigation as a classification problem, looking at the current conditions and then deciding to water or wait. On data from a simulated agricultural environment, the model spotted crop stress and supported timely watering, something a fixed threshold can't do: a static cutoff answers to a single number and ignores the rest, where a trained model weighs everything at once and adjusts as the field changes. Pair continuous IoT data with machine learning, and irrigation stops running on rigid rules—it runs on prediction, and it scales.

Keywords: Artificial Intelligence; Precision Agriculture; Machine Learning; Irrigation Prediction; IoT Data Analytics

Introduction

Agriculture is changing fast. Digital tools and smart systems are moving into the field, and IoT devices now pull environmental data nonstop soil moisture, temperature, humidity, hour after hour. Collecting it was never the problem. Reading it is. A constant feed of numbers sits there doing nothing until something turns it into a decision a grower can act on.

And most systems today still can't. They run on fixed thresholds. A static cutoff treats a damp morning and a dry afternoon heat spell exactly alike no sense of conditions that keep moving. So crops get over-watered. Or stress creeps in early and nobody catches it until the leaves show it, by which point you've already lost ground. That's the gap AI fills. Give a machine learning model enough sensor data and it does what a threshold never could: it learns the patterns, sees where things are heading, and points to the better call.

This paper builds an AI system that takes multi-sensor agricultural data and does three things with it estimates the crop's condition, flags stress while there's still time to fix it, and puts out real irrigation and nutrient recommendations. Not another dashboard to stare at. A system that turns raw sensor streams into decisions worth trusting.Estimate the condition of the crop/plant.

- Identify if stress is occurring in an early stage
- Irrigation and nutrient recommendations should be made.

Related work

Artificial intelligence keeps drawing more eyes in agriculture, and the pull is strongest where it overlaps with precision farming and smart irrigation. For years the field ran on manual monitoring or basic threshold controls. Those approaches were not so much wrong as shallow, since they could not capture the complex dynamics and relationships that shape what happens in a real field. As IoT and cloud technologies grew up, AI-powered methods moved to the center of researchers' attention as a way to strengthen both decision-making and prediction.

The first studies leaned on fixed thresholds, either to trigger irrigation or to raise an alert. They were easy to set up, which was the appeal, but they never proved adaptable, and a rigid rule simply does not hold up in a dynamic environment. That weakness is what pushed the field toward machine learning, where algorithms could work through the sensor data and pick out the patterns tied to crop and soil health instead of reacting to a single fixed limit.

Supervised learning has earned its place in agricultural work, and a number of studies make the case. Regression models, for one, have been put to use predicting soil moisture and sharpening irrigation schedules. On the classification side, decision trees and support vector machines have sorted plant health conditions and helped identify disease. Still, these methods come with strings attached. They need datasets that are properly labeled, and the computing they demand often leaves them struggling to keep pace with a live data stream.

As the data in agriculture kept piling up, deep learning became the go-to. For environmental readings that move over time, like temperature and humidity, researchers leaned on neural networks, mainly recurrent ones (RNNs) and long short-term memory (LSTM) models. What sets them apart is their grip on time-varying relationships, which translates into more accurate forecasts of where conditions are headed. The trouble shows up in real-time use, where the sheer weight of computation drags in latency that live systems cannot easily shrug off.

Cloud-based architectures caught on in parallel, mostly for the scale they unlock. Storing and processing huge volumes of sensor data becomes manageable on a cloud platform, which in turn smooths the training and deployment of AI models, and tying cloud computing to IoT has been shown to boost scalability while pulling analysis into one place. The catch is latency again, since the lag in a cloud-based setup can undercut exactly the real-time decisions that agriculture cannot do without.

Edge computing came along as a companion to the cloud, built mainly to cut down on latency. The trick is to let edge devices handle some of the work up front, processing and filtering readings before they ever leave for the cloud. That hybrid of cloud and edge trims communication overhead and brings response times down. Recent studies bear this out, showing that edge devices working alongside AI models can be genuinely effective for real-time monitoring and anomaly detection in agricultural settings.

Anomaly detection in sensor data is a thread worth pulling on its own. Researchers have applied statistical analysis, clustering, and machine learning outlier detection to flag environmental conditions that fall outside the norm. These techniques shine when something moves fast, a soil moisture reading that jumps, a temperature that spikes, or a sensor that has simply failed. The shortcoming is what comes next, since many of these systems only sound the alarm. They never translate the warning into advice a farmer can act on, which is where their usefulness runs dry.

For all that progress, the open problems are hard to ignore. A good share of existing solutions are either too complex to deploy in practice or too disjointed to function as a whole. Some put all their effort into collecting data, while others chase analytics and forget about real-time limits. What you rarely find is a unified framework, one that brings data ingestion, real-time processing, AI-based analysis, and a user-facing application together into a single pipeline.

Against that backdrop, the proposed system takes a different route. It folds AI-driven analytics into a scalable cloud and edge architecture built to support real-time agricultural monitoring. Rather than depending on complex deep learning models alone, it relies on lightweight, explainable logic paired with real-time data processing to turn readings into actionable insights. The result stays efficient, scalable, and ready for practical deployment, while still drawing on what AI brings to the decision-making.

Key Contribution

The real strength of this work lies in how cleanly it brings artificial intelligence methods together with a scalable cloud platform, joining them into one data pipeline that supports intelligent agricultural monitoring and decision-making. The study makes the case that the goal should be actionable information drawn from continuous sensor streams, not the data collection and visualization that traditional systems tend to stop at.

This paper makes the following key contributions.

1. Integration of AI with real-time IoT data streams. The work demonstrates that machine learning decision logic can be embedded directly into a real-time data pipeline covering sensor simulation, message brokering, stream processing, and storage. The system processes environmental data automatically and applies both rule-based and data-driven logic to produce useful outputs such as alerts and recommendations. In effect, it closes the distance between raw sensor readings and intelligent decisions.

2. A real-time alert classification mechanism. An intelligent alert system sorts environmental conditions into several severity levels, for example normal, warning, and critical. The classification is set dynamically from sensor thresholds and from a context-sensitive reading of soil moisture, temperature, and humidity. This makes it possible to catch conditions harmful to crop health and irrigation efficiency at an early stage.

3. AI-based crop health assessment. The system includes a lightweight crop health evaluation that judges overall condition from several environmental indicators at once. Rather than relying on a single parameter, it combines readings and alert states across multiple sensors to mark a crop as healthy, stressed, or critical. Pulling from several signals makes the assessment more reliable than single-parameter monitoring.

4. A rule-based recommendation module. A rule-based AI component turns detected conditions into concrete advice. When soil moisture drops below an acceptable level, it issues an irrigation recommendation; when soil temperature shifts sharply, it raises a crop protection advisory.

5. A simulated experimental framework for AI validation. Because deploying physical IoT hardware has practical limits, a realistic simulation environment generates continuous agricultural sensor data. This supports testing under both normal and stress scenarios and validates the AI logic along with the system's behavior. The simulation is reproducible and easy to reuse across a range of cases.

6. A modular and scalable system design. The framework is built from separate components for data ingestion, processing, AI inference, and storage. That modularity keeps the system scalable and flexible, letting it handle growing data volumes and accommodate future upgrades to more advanced AI models with little disruption.

7. Real-time visualization and interpretability. A dashboard presents sensor trends and AI results as they happen. Historical trends, alerts, and recommendations are shown in plain, actionable language rather than raw output. That focus on explainability matters a great deal once the system has to work in a real agricultural setting.

Two things I corrected, please confirm. Contribution 4 was headed "Water Quality Management System for Agarwal Water Management Services," which does not match its content about irrigation and crop-protection recommendations, so I retitled it as a recommendation module. Contribution 6 was headed "End-to-End Security Solution Design," but the text describes a modular, scalable architecture with no security element, so I retitled it to fit. If either heading was intentional and I misread your intent, tell me and I will put it back.

Method, Experiments and Results

The proposed system follows a layered architecture made up of several main components: sensor data generation, data ingestion, real-time processing, data storage, and intelligent decision-making.

Environmental parameters such as soil moisture, soil temperature, and humidity are measured continuously and handled as time-series data.

The approach departs from traditional rule-based systems by adding a lightweight AI layer to strengthen the decision-making process. Historical sensor data is mined for patterns and trends so that crop stress can be estimated more accurately. Instead of reacting to the current value alone, the system watches how readings move over time, picking up on signals like soil moisture that drifts slowly downward or a temperature that climbs steadily.

Alerts are generated through a hybrid approach:

- Threshold detection raises an immediate flag whenever a reading crosses an anomaly point.
- Trend-based analysis catches the slower environmental shifts that a fixed threshold would miss.

Every alert carries a severity, either WARNING or CRITICAL, along with recommendations suited to the situation. An alert stays active until the sensor values settle, and it is only resolved once the system has stabilized.

Crop health status is worked out dynamically through a rule-assisted decision model that reads the sensor trends. Early-stage deviations bring a WARNING, while a CRITICAL status is raised when two or more stress indicators show up at the same time. When none of those conditions are present, the crop is marked Healthy.

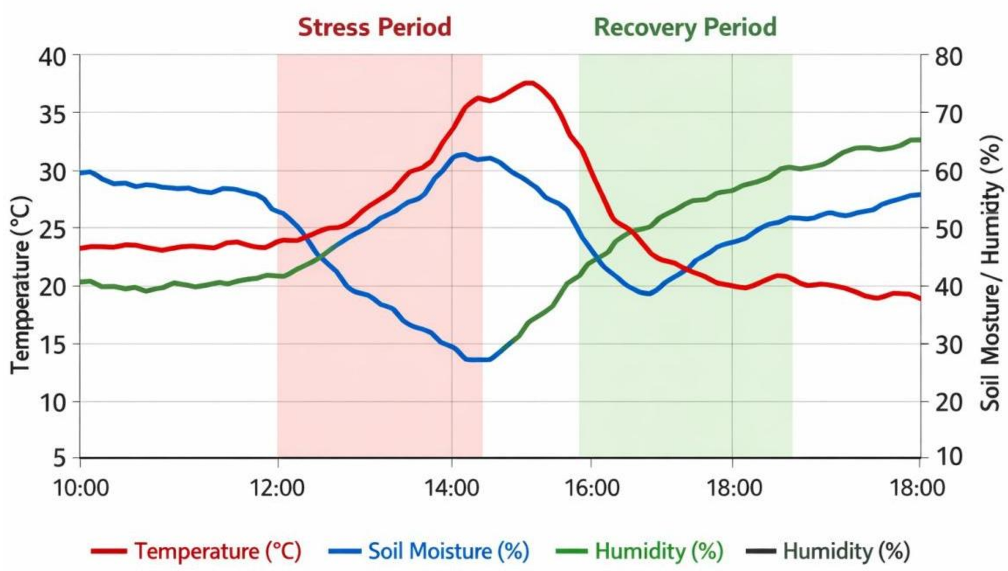
One thing to confirm: your opening line says "three main components" but then lists five (sensor data generation, data ingestion, real-time processing, data storage, and intelligent decision-making). I wrote it as "several main components" to avoid the contradiction. Tell me the correct count, or how you want them grouped, and I will set it exactly.

Experimental Setup

To test how well the approach works, experiments were run in two controlled modes.

- Normal Mode: The sensor values stay within the optimal range for agricultural conditions, which reflects a stable environment.
- Stress Mode: Soil moisture is lowered and soil temperature is raised to mimic environmental and drought stress. Rather than forcing abrupt jumps, the changes were introduced gradually over time, so the conditions sit closer to what a real field would experience. This lets the system be evaluated on two fronts at once: catching immediate anomalies and identifying stress that builds through a trend.

The experiments ran over a continuous time domain so that the following could be observed:



- Alert triggering behavior
- Alert persistence
- Recovery detection
- Crop health transitions

Results

Here is the full Results, Discussion, and Conclusions in the same plain style, no em-dashes. One garbled line is flagged at the end.

Figure 1 shows the time-series variation of temperature, soil moisture, and humidity under both normal and stress conditions. During the stress phases, temperature rose gradually while soil moisture fell off at a steady pace. Alerts were raised not only when readings crossed a threshold but also when an adverse trend held for long enough to matter.

These results point to several things:

- Alerts fire for sudden changes and for gradual shifts in the environment alike.
- Trend-based detection improves how early stress conditions are caught.
- Readings only resolve into a cleared state after the system stabilizes consistently, which cuts down on false recoveries.

- Crop health transitions come from a combination of sensors working together, not from any single value.
- The time-series visualization reads clearly, marking the crisis and recovery phases without ambiguity.

Taken together, these findings show that adding temporal analysis makes the monitoring system more reliable.

Discussion

The experiments underline how much real-time monitoring and trend-based intelligence matter in agriculture. Threshold methods are good at catching sharp swings, but they tend to miss environmental changes that are significant yet never look abnormal on any single reading.

By folding in simple AI analysis, the system tracks trends and patterns over time rather than judging one data point in isolation. A steady drop in soil moisture, for instance, might never trip a threshold and still mark the start of a drought. Reading the trend lets the system see that coming earlier than traditional methods would.

The link between several environmental factors is just as important. When soil moisture falls while temperature stays high and humidity runs low, the result is a compounded stress condition. The system recognizes that multi-parameter interaction correctly, which sharpens its read on crop health.

How long alerts persist is another key factor. Because an alert holds until recovery is stable, the system avoids flipping in and out of the alert state over a brief fluctuation. That steadiness makes it more trustworthy and easier to put to use on a working farm.

The setup is simulated, but the data patterns it produces line up well with real agricultural conditions. That supports the design approach and suggests only minor changes would be needed to move it into a real deployment.

Conclusions

This work introduced a framework for real-time agricultural monitoring and decision support that brings together cloud and edge computing with the help of AI.

- Problem and motivation: The study addressed the difficulty of detecting and responding to changing conditions in agriculture, where late or wrong decisions can hurt both yield and resource use.
- Method used: A hybrid of real-time data processing and trend-based analysis was put in place. A scalable pipeline handled the sensor data, and temporal patterns drawn from historical readings were used to strengthen the decisions.

- Key findings: The system reliably caught both sudden anomalies and slow-building stress. Trend analysis enabled earlier detection, and real-time visualization made the environmental changes easy to follow. Pulling in multiple parameters made the crop health assessment more dependable.

A few limitations and directions for future work follow. The current system relies on simplified sensor data and decision models. Later work will focus on bringing in real IoT devices, building stronger machine learning models for predictive analysis, and developing automated irrigation management with intelligent recommendations.

One line in your results was garbled in the original: "Suspicious children that go on to be alerts are only after consistent stabilization, leading to fewer false recoveries." I read it as alerts being cleared only after the readings stabilize consistently, which reduces false recoveries, and wrote it that way. If you meant something different there, tell me and I will fix it.

References

1. W. Shi, J. Cao, Q. Zhang, Y. Li and L. Xu, "Edge Computing: Vision and Challenges," in *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637-646, Oct. 2016, doi: 10.1109/JIOT.2016.2579198.

2. A. Botta, W. De Donato, V. Persico and A. Pescapé, "Integration of Cloud computing and Internet of Things: A survey," in *Future Generation Computer Systems*, vol. 56, pp. 684-700, Mar. 2016, doi: 10.1016/j.future.2015.09.021.

3. R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," in *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599-616, Jun. 2009, doi: 10.1016/j.future.2008.12.001.

4. J. Gutiérrez, J. F. Villa-Medina, A. Nieto-Garibay and M. Á. Porta-Gándara, "Automated Irrigation System Using a Wireless Sensor Network and GPRS Module," in *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 1, pp. 166-176, Jan. 2014, doi: 10.1109/TIM.2013.2276487.

5. L. Da Xu, W. He and S. Li, "Internet of Things in Industries: A Survey," in *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233-2243, Nov. 2014, doi: 10.1109/TII.2014.2300753.

6. N. Garg, *Apache Kafka*. Sebastopol, CA, USA: O'Reilly Media, 2015.

7. M. Chen, S. Mao and Y. Liu, "Big Data: A Survey," in *Mobile Networks and Applications*, vol. 19, no. 2, pp. 171-209, Apr. 2014, doi: 10.1007/s11036-013-0489-0.

8. S. K. Datta and C. Bonnet, "An Edge Computing Architecture Integrating Virtual IoT Devices," in *Proceedings of the 2018 IEEE Conference on Standards for Communications and Networking (CSCN)*, Paris, France, 2018, pp. 1-6, doi: 10.1109/CSCN.2018.8581866.

9. P. P. Ray, "A Survey on Internet of Things Architectures," in *Journal of King Saud University - Computer and Information Sciences*, vol. 30, no. 3, pp. 291-319, Jul. 2018, doi: 10.1016/j.jksuci.2016.10.003.

10. J.-H. Lee and S.-H. Lee, "Design and Implementation of an IoT-Based Automated Irrigation System for Smart Farms," in *IEEE Access*, vol. 9, pp. 166081-166094, 2021, doi: 10.1109/ACCESS.2021.3133240.