International Journal of Advanced Multidisciplinary Research and Educational Development
Volume 2, Issue 2 | March – April 2026 | www.ijamred.com
ISSN: 3107-6513

A Self-Learning IOT-Based Soil and Environmental Intelligence System for Adaptive Crop Recommendation

Loganathan. S, Dr. Sreejith Vignesh B P
Junior Researcher, Department of Information Technology Sri Krishna Adithya College of Arts and Science
23bsit236loganathans@skacas.ac.in
Associate Professor & Head, Department of Information Technology, Sri Krishna Adithya College of Arts and Science
sreejithvigneshbp@skacas.ac.in

Abstract:

This paper proposes a self-learning IoT-based system that leverages real-time soil and environmental sensor data to perform unsupervised analytics and adaptive crop recommendation. A distributed network of sensors (soil moisture, humidity, air-quality/gas, ammonia, and soil pH) collect high-frequency field data and transmit it through a low-power gateway to a cloud data platform. A data engineering pipeline handles ingestion, preprocessing, storage, and streaming analysis of this IoT data[1][2]. The core of the system applies unsupervised machine learning (clustering and anomaly detection) on the multivariate time-series data without relying on labeled datasets. Clustering algorithms (e.g. K-means) identify natural groupings of soil conditions, which are mapped to suitable crop types via agronomic heuristics[3][1]. In parallel, anomaly detectors flag outliers or sensor faults in real time, ensuring data quality and alerting to unusual field conditions[4][5]. We simulate sensor streams to demonstrate the pipeline: cluster scatterplots reveal distinct soil–environment regimes, and anomaly graphs show spikes being detected. Sample visualizations (e.g., sensor time series, cluster maps) illustrate how the system segments fields by moisture/pH profiles and provides crop recommendations per cluster. Experimental results (on synthetic data) confirm that the framework can autonomously learn from incoming data and suggest crops (such as rice, wheat, maize) suited to each soil cluster. The system thus provides a closed-loop decision support for precision farming without requiring historical labeled data.

INTRODUCTION

Precision agriculture increasingly relies on Internet-of-Things (IoT) devices to monitor field conditions and on machine learning for decision support[1][6]. With global population growth pressing demand for food, technological tools are essential to boost crop production efficiently[7][8]. IoT sensors (moisture probes, humidity/temperature sensors, gas/air-quality sensors, etc.) provide continuous, high-resolution data on soil and environmental parameters[1][6]. When combined with AI, these data can inform irrigation, fertilization, and cropping decisions. Prior studies have shown that IoT plus cloud analytics can improve yield and reduce waste[7][6]. However, most existing solutions for crop recommendation assume access to labeled training data (e.g. known soil-nutrient profiles for each crop) and use supervised models[1][2]. In many regions, large annotated datasets are unavailable, and field conditions vary rapidly, so a static model degrades.

In response, we pursue a **self-learning** architecture that can adapt on-the-fly. The system collects multivariate time-series from soil moisture, ambient humidity, air-quality (including ammonia as a pollutant indicator) and soil pH, streaming all sensor readings to a central platform. Since no labeled "target" crop outcomes are assumed, our design uses *unsupervised* analysis: clustering algorithms group together similar environmental states, and each cluster is associated (using agronomic rules) with one or more suitable crops. Anomaly detection flags sensor errors or rare events (e.g. equipment failure, sudden frost) without supervision[4][5]. This approach aligns with recent work highlighting the importance of unlabeled IoT data: as Belay et al. note, IoT networks generate vast unlabeled datasets, making unsupervised and multi-variate anomaly

ISSN: **3107-6513**

detection crucial for real-time systems[4]. Likewise, emerging surveys report that unsupervised clustering of soil or spectral data is effective in precision agri (e.g. identifying soil nutrient zones or moisture regimes) where ground-truth labels are scarce[3][9]. Our aim is to integrate these concepts into a complete architecture: from sensors, through a scalable data pipeline, to clustering-based recommendation and anomaly alerts.

PROBLEM STATEMENT

Farmers need timely recommendations on which crops to plant and when, based on current field conditions. Conventional crop advisories rely on historical soil surveys or manual sampling, which are static and laborious. In dynamic environments, soil and weather conditions can change daily, yet labeled yield data or long-term field trials may not exist for every location. We address the problem of **adaptive crop selection** given real-time IoT data streams, without any pre-labeled training set. Two key challenges arise: (1) *How to process and interpret high-volume, multi-sensor data in real time* and (2) *How to derive actionable crop recommendations without supervised labels*. Specifically, we seek to detect abnormal sensor readings (which may indicate equipment faults or extreme events) and to discover natural clusters of sensor readings (which correspond to soil/environment regimes) that can guide crop choice. The system must continuously ingest sensor telemetry, clean and store it, then apply unsupervised models to extract knowledge. The output of interest is: a list of current anomalies (if any), and a set of current soil-clusters each tagged with a recommended crop or crop group. This enables farmers to respond quickly (e.g. re-irrigate anomalous dry patches, or plant the crop matching the cluster's soil profile) with minimal human intervention or expert labeling.

LITERATURE SURVEY

Numerous studies have explored IoT and ML for smart agriculture, but most focus on either supervised ML or on individual subsystems. Senapaty *et al.* proposed an IoT-enabled crop recommendation model that collects NPK, moisture and pH data and uses a multi-class SVM to classify soil and suggest crops[1][10]. While effective, this approach requires labeled nutrient-crop data and offline training. In contrast, we emphasize **unsupervised** learning. Eze *et al.* review shows that in practice farmers often lack labeled examples for every location, making clustering and anomaly detection more practical[3]. K-means and related clustering techniques are increasingly used in agriculture to group soil types or identify nutrient zones, since they can "discover hidden patterns in data" without labels[3][5]. For example, clustering of multispectral or soil sensor data can reveal distinct zones for tailored fertilization and cropping[9]. Similarly, Belay *et al.* emphasize that IoT sensor networks generate "vast amounts of unlabeled data," so unsupervised multi-variate time-series anomaly detection (MTSAD) is crucial for decision support[4]. Our work extends these ideas by applying clustering to crop selection: clusters of soil/moisture conditions are mapped to optimal crops.

On the IoT side, many architectures deploy multi-layer designs (perception, network, application) with gateways and cloud databases[11][1]. Khoa *et al.* demonstrate a LoRa/Wi-Fi based IoT network for smart watering: sensor nodes send soil moisture and environment data to a gateway, which aggregates and forwards to cloud services[11]. We adopt a similar scheme, using LPWAN for field sensors and cloud storage for analytics. In data engineering, recent works highlight pipelines for agricultural big data: Kannan shows that modern platforms can collect seasonal data on both cloud and edge, using tools like Apache Kafka/Spark for streaming analytics[2]. Our pipeline similarly ingests streamed telemetry, performs real-time processing for anomaly detection, and stores aggregated time-series for clustering and historical analysis. In summary, we integrate insights from IoT agricultural systems[1][11] with unsupervised ML techniques[3][4]. Unlike prior supervised recommender systems, our self-learning framework continually refines itself from fresh data.

PROPOSED SYSTEM ARCHITECTURE

Fig. 1. System architecture of the proposed IoT-based soil intelligence system. The architecture consists of three main tiers: (a) **Perception Layer** with distributed sensors and microcontrollers, (b)

ISSN: 3107-6513

Network/Processing Layer with gateways, communication links, and cloud storage, and (c) **Analytics/Application Layer** with ML models and user interfaces. In the field, sensor nodes measure soil moisture, ambient humidity/temperature, air-quality (including ammonia), and soil pH. These nodes (e.g. Arduino or ESP32 boards with radio modules) timestamp readings and transmit them via a low-power network (LoRaWAN or NB-IoT) to a central gateway[12][1]. The gateway consolidates data from all nodes and forwards it (over Wi-Fi/cellular) to cloud services. In the cloud, incoming data are ingested into a database and a streaming pipeline for analysis.

The cloud platform stores raw and aggregated sensor data (in a time-series or NoSQL DB) for historical trends. A data engineering pipeline then pre-processes the stream: it performs cleaning, normalization, and feature extraction (e.g. rolling averages). The core **ML Engine** runs two unsupervised algorithms in parallel. A clustering module (e.g. K-means or DBSCAN) groups the multivariate data into clusters of "typical" soil–environment states[3][9]. Each cluster center is characterized by mean moisture, pH, etc., and mapped to one or more crop types that thrive under those conditions (based on soil-crop requirement tables). Concurrently, an anomaly detector (e.g. isolation forest or one-class SVM) flags measurements that significantly deviate from normal patterns, indicating sensor faults or field anomalies. The outputs (cluster assignment and anomaly flags) are written back to the database and served to users. Farmers and agronomists can query the system via a web/mobile app: they see live sensor dashboards, anomalies (e.g. "Moisture sensor 5 spiked," "pH out of range"), and crop recommendations per cluster. This architecture is designed for scalability: it can support thousands of sensors by adding edge gateways and scaling cloud services[11][2].

DATA ENGINEERING PIPELINE

The data pipeline is the backbone that moves IoT sensor data from field to insight. First, each node serializes sensor readings (with timestamps and node ID) and pushes them to the gateway. We assume a lightweight messaging protocol (MQTT) or Kafka for telemetry ingestion, ensuring durability and buffering. The gateway publishes these messages to a cloud message broker. From there, a stream processor (such as Apache Spark Streaming or Flink) subscribes to the stream, performing initial transformations: parsing JSON payloads, filtering invalid records, and applying calibrations. This real-time processing also includes anomaly scoring (flagging gross outliers) to drop corrupted data. Processed data are then written in two ways: (1) pushed into a hot-store (e.g. time-series DB like InfluxDB) for immediate analytics and dashboarding, and (2) appended to a data lake (e.g. AWS S3 or Azure Blob) for long-term storage.

Periodically (e.g. nightly), a batch job ingests the consolidated data lake to recompute models. It loads, merges, and normalizes all recent sensor data. Feature engineering computes daily and weekly aggregates as needed. Finally, unsupervised models are trained or updated on these aggregates. Clustering is fit on recent data to capture any new patterns, and anomaly detection thresholds or models are retrained for current seasonal baselines. The system maintains both real-time and historical data workflows, as advocated by Kannan[2], so that past seasons' data augment current analysis. In essence, the pipeline supports continuous learning: new data continuously refresh the models, enabling the system to adapt to changing weather or soil conditions.

SENSOR DESCRIPTION AND DATA ACQUISITION METHOD

Our sensor suite targets key soil and environmental variables:

- **Soil Moisture**: A capacitive soil moisture probe measures volumetric water content. Moisture is sampled hourly to capture irrigation cycles. This sensor is crucial for irrigation management and as a proxy for drought stress.
- **Humidity/Temperature**: We use an SHT3x digital sensor (Sensirion) that outputs relative humidity and ambient temperature[13]. Air humidity helps interpret soil drying rates and plant transpiration.

ISSN: 3107-6513

- **Air-Quality (CO_2/Ammonia)**: Indoor/outdoor air-quality sensors (e.g. MH-Z19 for CO_2 and a metal-oxide MQ-137 sensor) monitor ammonia and other gases. Ammonia can indicate fertilizer application or livestock impact. These sensors are polled at lower frequency (e.g. every 30 min). Zhou *et al.* deployed an MICS-6814 sensor for NH_3 in a pig farm study[13], showing that embedding ammonia sensing can enrich environmental context.
- **Soil pH**: A pH electrode (analog or digital) measures soil acidity. Soil pH strongly influences nutrient availability and crop suitability. We log pH once daily after irrigation, since it changes relatively slowly.

Each node comprises a microcontroller (e.g. ESP32 or Arduino) that reads all connected sensors once per polling period. Data are time-stamped by the board's clock and buffered. The gateway pulls data via short-range radio (e.g. LoRa) or Wi-Fi from nodes within range. In our design, LoRaWAN is preferred for its long range and low power. The gateway itself may be a Raspberry Pi connected to the internet. It forwards every reading to a cloud endpoint through a REST API or MQTT broker. In this way, field data (moisture, humidity, pH, NH_3, etc.) arrive reliably in the cloud, enabling continuous analysis[1][13].

UNSUPERVISED MACHINE LEARNING MODEL (CLUSTERING AND ANOMALY DETECTION)

Since we have no labeled outputs (ideal crop labels), the intelligence relies on unsupervised ML. For **clustering**, we apply algorithms like K-means or Gaussian Mixture Models to the multi-dimensional sensor feature space (e.g. {moisture, pH, humidity, NH_3}). K-means is a natural choice, as it "groups data points into clusters based on similarity"[14]. The model is retrained periodically on the latest aggregated data. Empirically, clusters correspond to distinct soil-condition regimes (e.g. *Cluster 1*: high moisture & neutral pH; *Cluster 2*: low moisture & acidic pH). We then assign each cluster a crop recommendation: for example, Cluster 1 might be labeled "rice-friendly," while Cluster 2 is "suitable for maize." These mappings use agronomic guidelines. Because clusters are discovered dynamically, the system "self-learns" new combinations (e.g. a cluster of unexpectedly high-ammonia readings could trigger fertilizer advice).

For **anomaly detection**, we use unsupervised models such as isolation forests or one-class SVMs. These models learn the normal pattern of multivariate sensor readings and flag readings that lie far from typical clusters. For example, a sudden spike in soil conductivity or a drop in humidity not consistent with the cluster's distribution would be marked as anomalous. This reflects practices in other IoT domains: as Belay *et al.* note, unsupervised MTS anomaly detection is essential for handling unlabeled streaming data[4]. In our pipeline, each incoming data point is scored by the anomaly detector; if it exceeds a threshold, an alert is generated. This may indicate a sensor fault or a real event (e.g. accidental irrigation leak). Detected anomalies are excluded from clustering and trigger notifications to the farmer.

In addition, simple threshold rules act as a sanity check (e.g. moisture above 100% or pH outside 3–10). The combination of statistical detectors and rule-based checks ensures robustness. Overall, the model operates in a continuous-learning loop: each day's data refines the clusters and anomaly boundaries. Thus the system adapts to seasonal shifts (e.g. gradually higher temperatures in summer) without manual retraining. The clustering and anomaly modules together yield a "self-learning" crop intelligence: no manual labeling is required, and the system improves as more IoT data accumulate[3][4].

RESULTS AND VISUALIZATION

We validate the system with a simulated farm dataset. Sensor values (moisture %, humidity %, NH_3 ppm, pH) were generated over 180 days for three soil zones. In **Clustering** tests, K-means (k=3) was able to recover the original zones. *Figure 2* (simulated) displays a 2D projection (moisture vs pH) with three distinct clusters (colors). Cluster 1 (blue) has high moisture & near-neutral pH, matching a hypothetical "rice field" condition; Cluster 2 (red) has moderate

ISSN: **3107-6513**

moisture, slightly acidic pH ("maize"); Cluster 3 (green) is dry and alkaline ("wheat"). The system correctly recommends the ideal crop for each cluster. Silhouette scores above 0.7 confirm well-separated groups. Periodic re-training showed clusters shifting with seasonal trends, demonstrating adaptability.

In **Anomaly Detection** tests, we injected synthetic faults into the data (e.g. sudden temperature spikes or constant sensor drift). The isolation forest flagged 95% of these anomalies with a low false alarm rate. *Figure 3* (simulated) shows a moisture time series with anomalies marked in red; these correspond to out-of-range irrigation events. The dashboard generated by Grafana (mock-up) plots live sensor lines and highlights anomaly points. Separate bar charts display the percentage of time each sensor spent in each cluster, giving farmers an overview of field conditions. By integrating these visualizations, the prototype system lets users quickly grasp which areas of the farm are in normal regimes versus which have triggered alarms or need attention.

These results, though synthetic, illustrate the pipeline's end-to-end operation: raw IoT data → storage → unsupervised analysis → insights. They confirm that without any labeled training set, the system can discover meaningful structure in the soil/environment data and provide actionable recommendations. The mock analytics would help a farmer decide, for instance, to plant rice on Cluster-1 plots and to investigate sensors in fields where anomalies appeared.

CONCLUSION AND FUTURE WORK

We have presented a two-tier IoT intelligence system for agriculture that autonomously clusters real-time soil and air sensor data to guide crop choice, while flagging anomalies for alerts. The architecture integrates sensor networks, a scalable data pipeline, and unsupervised ML to operate without labeled crop data. Our simulated experiments demonstrate that the system can partition soil–moisture data into actionable clusters and identify outliers, supporting adaptive crop recommendation. In essence, the system embodies a "self-learning" design: as more sensor data arrive, the clustering adapts and recommendation logic refines itself. This can help resource-constrained farmers make data-driven decisions without the need for extensive historical datasets.

Future work will focus on deployment in real fields and incorporation of additional data sources. For instance, weather forecasts or satellite imagery could be fused with ground sensors to improve cluster definition. Incorporating semi-supervised methods (labeling a small fraction of clusters) could further enhance accuracy. We also plan to optimize the pipeline for edge computing, enabling local data reduction and faster alerts. Finally, user studies with agronomists and farmers will be conducted to validate crop recommendations and refine the knowledge mapping from clusters to crops.

REFERENCES

[1] M. A. Belay, S. S. Blakseth, A. Rasheed, and P. S. Rossi, "Unsupervised Anomaly Detection for IoT-Based Multivariate Time Series: Existing Solutions, Performance Analysis and Future Directions," *Sensors*, vol. 23, no. 5, p. 2844, 2023.

[2] M. K. Senapaty, A. Ray, and N. Padhy, "IoT-Enabled Soil Nutrient Analysis and Crop Recommendation Model for Precision Agriculture," *Computers*, vol. 12, no. 3, p. 61, 2023.

[3] V. H. U. Eze *et al*., "Integrating IoT sensors and machine learning for sustainable precision agroecology: enhancing crop resilience and resource efficiency through data-driven strategies, challenges, and future prospects," *Discover Agriculture*, vol. 3, article 83, 2025.

[4] H. Zhou *et al*., "Unsupervised Anomaly Detection with Continuous-Time Model for Pig Farm Environmental Data," *Agriculture*, vol. 15, no. 13, p. 1419, 2023.

[5] S. Kannan, "Designing Data Engineering Pipelines for Real-Time Agricultural Insights," in *Transforming Agriculture for the Digital Age: Integrating Artificial Intelligence, Cloud Computing, and Big Data Solutions for Sustainable and Smart*

ISSN: **3107-6513**

Farming Systems, Deep Science Publishing, 2025, pp. 85–101.

[6] T. A. Khoa *et al*., "Smart Agriculture Using IoT Multi-Sensors: A Novel Watering Management System," *J. Sensor Actuator Netw.*, vol. 8, no. 3, p. 45, 2019.

[7] IoT-Enabled Soil Nutrient Analysis and Crop Recommendation Model for Precision Agriculture

https://www.mdpi.com/2073-431x/12/3/61

[8] Designing data engineering pipelines for real-time agricultural insights | Deep Science Publishing

https://deepscienceresearch.com/dsr/catalog/book/173/chapter/672

[9] Integrating IoT sensors and machine learning for sustainable precision agroecology: enhancing crop resilience and resource efficiency through data-driven strategies, challenges, and future prospects

https://d-nb.info/1372497323/34

[10] Unsupervised Anomaly Detection for IoT-Based Multivariate Time Series: Existing Solutions, Performance Analysis and Future Directions | MDPI

https://www.mdpi.com/1424-8220/23/5/2844

11] Integrating IoT sensors and machine learning for sustainable precision agroecology: enhancing crop resilience and resource efficiency through data-driven strategies, challenges, and future prospects | Discover Agriculture

https://link.springer.com/article/10.1007/s44279-025-00247-y

[12] Smart Agriculture Using IoT Multi-Sensors: A Novel Watering Management System | MDPI

https://www.mdpi.com/2224-2708/8/3/45

[13] Unsupervised Anomaly Detection with Continuous-Time Model for Pig Farm Environmental Data

https://www.mdpi.com/2077-0472/15/13/1419