ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26nov646

A Hybrid 1D CNN-LSTM Model for Anomaly Detection in Agricultural IoT Systems

Mustapha Abubakar Ibrahim[1]; Nurudeen Ibrahim[2]; Dr. Ridwan Kolapo[3]; Prema Kirubakaran[4]

[2]Associate Professor, [4]Professor

[1,2,3,4]Department of Information Technology, Faculty of Computing Nile University of Nigeria Abuja, Nigeria

Publication Date: 2026/11/20

Abstract: Smart farming depends on Internet of Things (IoT) sensors that report temperature, humidity, soil moisture and light intensity around the clock. When these sensors fail or their readings are corrupted, automated decisions such as irrigation and climate control can waste resources and reduce yield. Threshold rules and classical machine learning methods struggle with the daily and seasonal cycles found in farm data, and they often raise too many false alarms. This study develops and tests a hybrid 1D Convolutional Neural Network and Long Short Term Memory (CNN-LSTM) model for anomaly detection in agricultural IoT systems. The model was trained on a real dataset of 233,980 sensor readings with four features. Preprocessing covered missing value interpolation, normalization, sliding window sequence creation, feature engineering and feature selection by recursive elimination. Convolutional layers extract local patterns across sensors, and the LSTM layer models how those patterns change over time. On the held out test data the model reached an accuracy of 91.08 percent, a precision of 100 percent, a recall of 82.16 percent, an F1 score of 90.20 percent and an area under the curve of 91.06 percent. It produced no false positives, which matters on farms where false alarms lead to needless interventions and lost trust. Comparison with results from earlier studies shows that the model gives the highest precision while staying competitive on overall detection. The work offers a dependable approach to fault detection in smart farming that favors precision, and future studies should look at edge deployment, interpretability and adaptive learning.

Keywords: *Anomaly Detection; Agricultural IoT; CNN-LSTM; Deep Learning; Smart Farming; Time Series Data.*

How to Cite: Mustapha Abubakar Ibrahim; Nurudeen Ibrahim; Dr. Ridwan Kolapo; Prema Kirubakaran (2026) A Hybrid 1D CNN-LSTM Model for Anomaly Detection in Agricultural IoT Systems. *International Journal of Innovative Science and Research Technology*, 11(11), 849-854. https://doi.org/10.38124/ijisrt/26nov646

I. INTRODUCTION

Agriculture remains a basis for food security and employment in many countries, and Nigeria is a clear example [1]. Even so, traditional farming methods struggle with climate change, water shortages and the wasteful use of inputs [2]. These pressures have pushed researchers and farmers toward smart farming, also called precision agriculture. In this approach, sensors placed across the field record soil moisture, temperature, humidity and light as they change, and the readings are sent to systems that help farmers decide when to irrigate, what nutrients to apply and how to protect their crops [3].

A system of this kind is only as good as the data it receives. Sensors can break, wireless links can drop, bad weather can disturb the readings and, in some cases, attackers can interfere with devices. A soil moisture sensor that wrongly reports dry soil, for instance, may trigger irrigation that is not needed and harms the crop. One study estimates that between 15 and 30 percent of sensor data can be inaccurate, which leads to poor decisions and financial losses [4].

Many detection methods still depend on fixed thresholds or simple statistical rules [5]. These rules ignore the fact that farm data follows daily and seasonal cycles. A reading of 40 degrees Celsius might be ordinary in the afternoon and suspicious at night [6]. As a result, such systems miss real faults or raise too many false alarms, and farmers soon stop trusting them. Machine learning and deep learning offer a better route because they learn normal behavior directly from past data [7]. Among deep models, hybrids of convolutional neural networks (CNNs) and Long Short Term Memory (LSTM) networks have shown strong results [8]. The CNN part is good at spotting local patterns such as sudden spikes or drops, and the LSTM part keeps track of longer term context.

Most detectors in the literature were designed for industrial or general IoT settings [9]. They pay little attention to features that are common on farms, including a heavy

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26nov646

imbalance between normal and faulty readings, different sensor types and positions, and clear daily and seasonal patterns [10]. This paper addresses that gap with a 1D CNN-LSTM model built for agricultural sensor data. The work makes three contributions. It sets out a preprocessing and feature engineering pipeline suited to farm sensor streams. It designs and trains a hybrid model that captures both local sensor patterns and temporal dependencies. It evaluates the model with accuracy, precision, recall, F1 score and area under the curve (AUC), and compares the results with earlier studies.

II. RELATED WORK

➢ *Anomaly Detection in Agricultural IoT*

Anomalies in sensor data are departures from expected behavior. They can come from device failure, power loss, communication problems, extreme weather or deliberate attacks [11]. If they go unnoticed, automated systems may act wrongly, for example by irrigating during a rainstorm or missing a frost event [12]. Supervised methods need labeled examples of both normal and faulty data, which are hard to get on farms. Unsupervised and semi supervised methods instead learn what normal data looks like and flag whatever departs from it [13].

Deep learning has changed this area. LSTM networks capture long range dependencies and suit data with daily or seasonal cycles [14], while CNNs find local patterns and relationships across several sensor channels over short periods [15]. Putting the two together lets one model look at both the immediate shape of a signal and its longer context.

➢ *Machine Learning in Agriculture*

Machine learning is now used for early disease detection [16], [17], yield prediction [18], [19] and resource optimization [20], [21]. As sensor networks grow and sampling rates rise, datasets become larger and richer in time, which favors deep learning over classical methods [22], [23]. Hybrid architectures are becoming common because one framework can handle both feature extraction and temporal modeling [24].

➢ *Earlier Studies on Anomaly Detection*

Rashid et al. [25] applied a Random Forest classifier to soil moisture data and reached an accuracy of 89 percent, but the model often mistook natural wetting after rain for sensor failure. This shows what happens when a model has no sense of time. Wang et al. [26] used an autoencoder to monitor greenhouse climate and trained it only on normal data. It detected sudden faults such as heater or ventilation failures, but it was sensitive to small sensor noise and needed a large amount of clean training data. Autoencoders flag readings with a high reconstruction error [27], and their sensitivity to noisy training data has been noted elsewhere [28].

Liu et al. [29] used an LSTM to detect faults in automated irrigation. Recall was high at 92 percent but precision was only 85 percent, so many alarms were false. A hybrid CNN-LSTM raised precision to 90 percent while keeping recall high. Outside agriculture, Zhao et al. [30] built a 1D CNN-LSTM for fault diagnosis in industrial systems and obtained an F1 score of 94 percent, better than a CNN or an LSTM alone. A wider comparison of anomaly detection methods also placed hybrid CNN-LSTM models among the best performers across several datasets [31].

➢ *Research Gap*

Three gaps stand out. First, earlier agricultural studies often test single architectures such as an LSTM or an autoencoder instead of hybrids that combine complementary strengths. Hybrids work well in industrial and general time series tasks, but their careful adaptation to farm data has received less attention. Second, many studies rely on curated or simulated data that lack the noise, missing values and seasonal drift of real sensor networks, so reported performance may not hold in the field. Third, many deep models act as black boxes and give no reason for a flagged reading, which makes farmers less willing to trust them. This study responds mainly to the first two gaps by training a hybrid model on real farm sensor data. Interpretability is left for future work.

III. METHODOLOGY

➢ *Framework Overview*

Fig 1 shows the framework. The sensor dataset is loaded and preprocessed and then passed to the 1D CNN-LSTM model. The data are split into 80 percent for training and 20 percent for testing, and the trained model is validated and evaluated before it produces anomaly predictions.

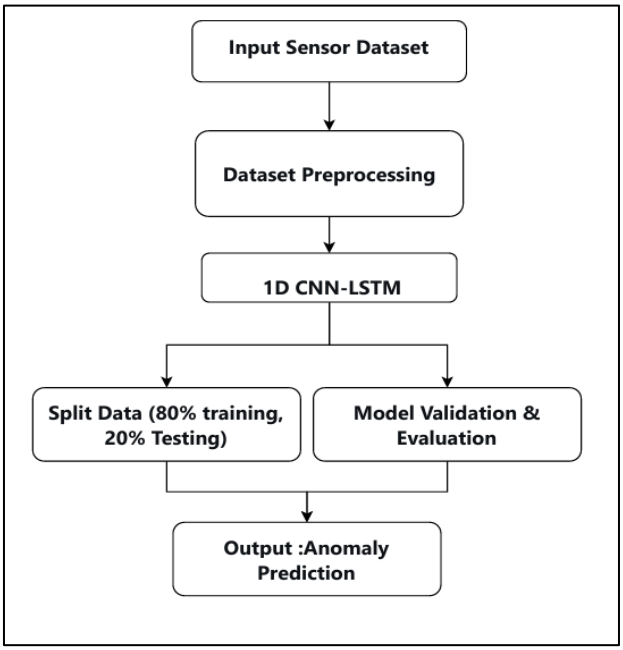


Fig 1 Research Framework of the Proposed System

➢ *Dataset*

The study uses a publicly available agricultural IoT dataset from the Kaggle repository. It holds 233,980 multivariate, time stamped readings from four sensors, namely air temperature (degrees Celsius), relative humidity (percent), soil moisture (cubic meters per cubic meter) and light intensity (lux). About 15 percent of the readings are anomalous, which reflects the situation on real farms where faults are important but uncommon. The data were loaded and handled with the Pandas library in Python.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26nov646

➢ *Data Preprocessing*

Gaps caused by transmission errors or sensor failures were filled by linear interpolation, which keeps the series continuous. A missing value at time t is replaced by the mean of the nearest valid readings before and after it:

$$x_t = (x_{t-1} + x_{t+1}) / 2 \quad (1)$$

This assumes that neighboring values change in a straight line, which is reasonable for slowly changing environmental variables over short intervals. Missing values are rare in the chosen dataset.

Each feature was then scaled to the range 0 to 1 using the minimum and maximum values of the training set, so that features with large magnitudes do not dominate learning:

$$x_{\text{norm}} = (x - x_{\min}) / (x_{\max} - x_{\min}) \quad (2)$$

The continuous series was cut into overlapping windows of fixed length L, moved across the normalized data with a fixed stride:

$$W_i = [x_i, x_{i+1}, \ldots, x_{i+L-1}] \quad (3)$$

Here x_t is the vector of all sensor features at time t, so every window has dimensions L by F, where F is the number of features. A window length of 24 time steps was used.

In addition to the raw readings, derived features were computed for each window. These include rolling means, rolling standard deviations, differences between successive time steps, and cyclical encodings of the hour of day and the day of year. They help the model pick up periodicity and variation. To avoid too many inputs and to reduce noise, recursive feature elimination with cross validation (RFECV) was applied. It repeatedly removes the least useful feature according to a baseline Random Forest classifier and keeps the subset that gives the best F1 score on the validation folds.

➢ *Model Architecture*

The model was built in Keras with a TensorFlow backend, and Table 1 summarizes it. The input is a tensor of shape L by F. Two 1D convolutional layers come first, one with 64 filters and one with 32, both with kernel size 3 and ReLU activation. They run along the time axis and learn local patterns and correlations between sensors within short segments of a window. The operation in layer l is

$$h^{(l)} = \text{ReLU}(W * h^{(l-1)} + b) \quad (4)$$

Where $*$ is the 1D convolution, W and b are learnable weights and biases, and $h^{(l-1)}$ is the output of the previous layer.

Table 1 Architecture and Training Settings of the 1D CNN-LSTM Model

Component	Setting
Input	24 time steps by F features
Conv1D layer 1	64 filters, kernel size 3, ReLU
Conv1D layer 2	32 filters, kernel size 3, ReLU
Max pooling	Pool size 2
LSTM layer	50 units
Dropout	Rate of 0.5
Output layer	1 neuron, sigmoid activation
Optimizer	Adam, learning rate 0.001
Loss and batch	Binary cross entropy, batch size 32
Early stopping	Patience of 10 epochs on validation loss

A max pooling layer with pool size 2 follows the convolutions. It halves the sequence length, reduces computation and keeps the strongest activations. The pooled feature maps go to an LSTM layer of 50 units, which updates a hidden state and a cell state at each time step. Its gates let it remember or forget information over long spans, so it can judge whether the local features form a pattern consistent with normal operation. A dropout layer with a rate of 0.5 comes next to reduce overfitting. The final layer is a dense layer with one neuron and sigmoid activation, which gives the estimated probability that a window contains an anomaly. The probability is turned into a class label with a decision threshold, which was tuned on validation data to 0.1628.

➢ *Training and Evaluation Setup*

The model was compiled with the Adam optimizer at a learning rate of 0.001 and trained with the binary cross entropy loss.

$$L = -(1/N) \Sigma [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (5)$$

Where y_i is the true label and $\hat{y}_i$ is the predicted probability. Training used mini batches of 32 windows. Early stopping watched the validation loss and ended training when it failed to improve for ten consecutive epochs, after which the weights from the best epoch were restored.

The data were split into training and testing sets of 80 and 20 percent with stratification, so that the share of anomalies stayed the same in both. Performance was measured with accuracy, precision, recall, F1 score and AUC, the area under the receiver operating characteristic curve. The first four follow from the counts of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN):

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad (6)$$

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26nov646

$$\text{Precision} = TP / (TP + FP) \quad (7)$$

$$\text{Recall} = TP / (TP + FN) \quad (8)$$

$$F1 = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (9)$$

Precision shows how many flagged readings are real anomalies and so reflects the false alarm rate. Recall shows how many real anomalies are found. AUC measures how well the model ranks anomalous cases above normal ones across all thresholds. Confusion matrices and training curves were also produced.

All experiments ran on Google Colaboratory with Python 3.10 or later, an NVIDIA Tesla T4 GPU, TensorFlow 2 with the Keras API, and the NumPy, Pandas, Scikit-learn, Matplotlib and Seaborn libraries. Training took about three hours.

IV. RESULTS AND DISCUSSIONS

➢ *Training Behavior*

Fig 2 shows the training history. The training loss fell from about 1.43 in the first epoch to about 1.37 by epoch 15, while the validation loss stayed almost constant near 1.38. The validation AUC stayed close to 0.911 throughout. The curves show stable training with no growing gap between training and validation loss. The flat validation curves also suggest that most of the learning happened early, so further tuning could still bring gains.

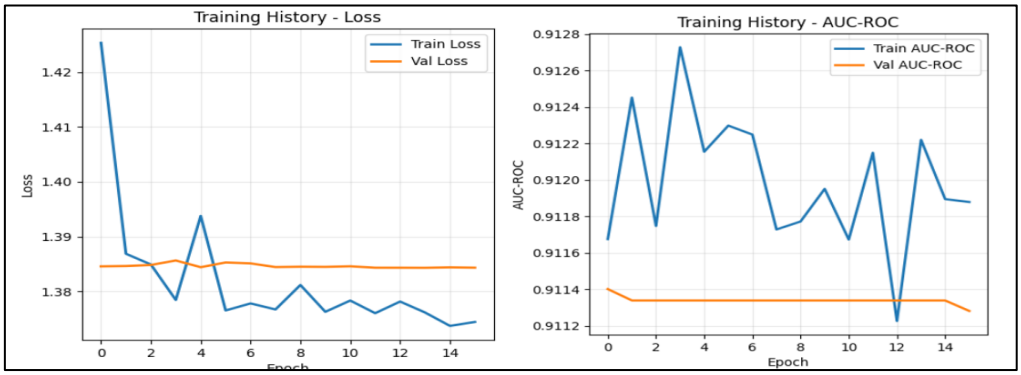


Fig 2 Training and Validation Loss (Left) and AUC (Right) Over the Training Epochs

➢ *Test Performance*

Table 2 gives the results on the held out test set. The model reached an accuracy of 0.9108, a precision of 1.0000, a recall of 0.8216, an F1 score of 0.9020 and an AUC of 0.9106. The confusion matrix in Fig 3 gives more detail. The model classified 39,772 normal samples and 32,678 anomalous samples correctly. It produced no false positives and missed 7,097 anomalies.

Table 2 Performance of the Proposed Model on the Test Set

Metric	Score	Interpretation
Accuracy	0.9108	High overall correctness
Precision	1.0000	No false alarms, every flagged anomaly is real
Recall	0.8216	About 82 percent of actual anomalies detected
F1 score	0.9020	Good balance of precision and recall
AUC	0.9106	Strong ranking ability across thresholds

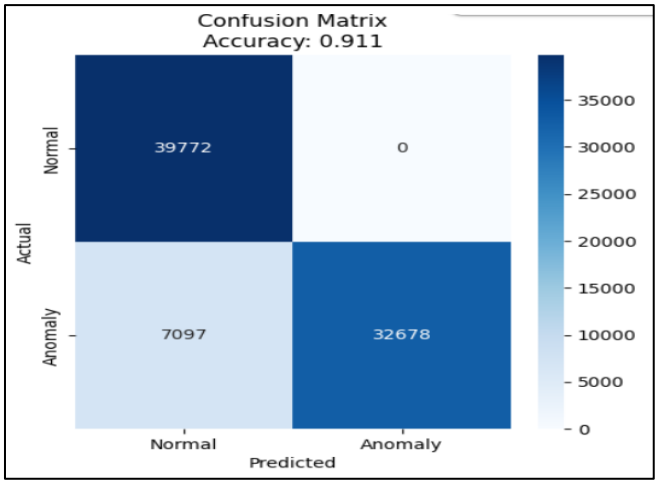


Fig 3 Confusion Matrix of the Proposed Model on the Test Set

A precision of 100 percent matters on farms. Every alert the system raises points to a real problem, so farmers are not sent to check healthy equipment and irrigation is not switched on or off without reason. The cost is a recall of 82.16 percent, which means that about 18 percent of anomalies go undetected. This reflects a deliberate preference for operational reliability over exhaustive detection. Where missed faults are more costly than false alarms, the decision threshold can be adjusted to move this balance.

➢ *Comparison with Earlier Studies*

Table 3 places the proposed model beside results reported in earlier studies. Those studies used different datasets and settings, so the comparison is a rough guide and not a head to head test.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26nov646

Table 3 Comparison with Results Reported in Earlier Studies

Model	Accuracy	Precision	Recall	F1 score
Random Forest [25]	0.8900	0.8720	0.8810	0.8760
Autoencoder [26]	0.9150	0.8980	0.9020	0.9000
LSTM [29]	0.9080	0.8500	0.9200	0.8830
CNN-LSTM, industrial [30]	0.9420	0.9380	0.9350	0.9360
Proposed 1D CNN-LSTM	**0.9108**	**1.0000**	**0.8216**	**0.9020**

The proposed model has the highest precision of all, at 1.0000, with no false positives. Its accuracy of 0.9108 is above the Random Forest result of 0.8900 and close to the autoencoder result of 0.9150. Its F1 score of 0.9020 is also slightly above those of the Random Forest, the autoencoder and the LSTM. The industrial CNN-LSTM of Zhao et al. scores higher on accuracy, recall and F1 score, but it was built for a different domain and does not reach the same precision. The recall of 0.8216 is lower than that of several other models, such as the LSTM of Liu et al. at 0.9200, and this is the price paid for removing false alarms. Overall, the F1 score shows a balanced result that suits settings where both false alarms and missed faults carry a cost, and the AUC of 0.9106 confirms that the model separates normal from anomalous sequences well across thresholds.

➢ *Limitations*

Several limits should be kept in mind. The model was tested on one public dataset, so its behavior on other crops, regions and sensor types is not yet known. Its recall is lower than that of some alternatives. It also gives no explanation of why a window was flagged. Finally, the comparison with earlier work relies on published figures that come from different data.

V. CONCLUSION AND RECOMMENDATIONS

This paper developed and evaluated a hybrid 1D CNN-LSTM model for anomaly detection in agricultural IoT systems. Convolutional layers extract local patterns across sensors, and an LSTM layer models how those patterns evolve in time. On a real dataset of 233,980 sensor readings the model reached an accuracy of 91.08 percent, a precision of 100 percent, a recall of 82.16 percent, an F1 score of 90.20 percent and an AUC of 91.06 percent, with no false positives. Compared with earlier studies, it gives the best precision while staying competitive overall. Because its alerts can be trusted, it can support timely action against crop loss, equipment failure and wasted resources.

Based on these findings, five recommendations are made. The model should be optimized and deployed on edge devices or farm gateways for low latency detection that does not depend on the cloud. It should be integrated with existing farm management software so that warnings appear inside daily workflows. Continuous learning would let it adapt to new sensors and seasons without full retraining. Adding other data, such as pH, nutrient levels and drone imagery, would give a fuller picture of farm health. Finally, attention mechanisms or visualization tools should be added so that users can see which sensor or time period caused a flag.

Future studies could test transfer of the model across crop types and regions, develop adaptive thresholds that follow operational priorities and seasonal trends, combine the data driven approach with physical models of soil and crop growth, run field trials over several growing seasons, and apply pruning, quantization or distillation so the model can run on low power IoT nodes.

ACKNOWLEDGMENT

The authors thank the Department of Information Technology, Faculty of Computing, Nile University of Nigeria, Abuja, for the support given during this research.

REFERENCES

[1.] Muhammed et al., “The role of agriculture in the economic diversification of the Nigerian economy (1980 to 2016),” Central Bank of Nigeria, 2018.
[2.] The Associated Press, “Nigeria has a food security problem as water for crops is harder to find,” 2025.
[3.] Jiang et al., “Farm-LightSeek: An edge-centric multimodal agricultural IoT data analytics framework with lightweight LLMs,” IEEE, 2025, doi: 10.1109/MIOT.2025.3575887.
[4.] Dabrowski et al., “Quality control in weather monitoring with dynamic linear models,” arXiv preprint arXiv:2211.04528, 2022.
[5.] Li et al., “Anomaly detection via self-organizing map,” arXiv preprint arXiv:2107.09903, 2021.
[6.] Koliopanos et al., “Enhancing temperature data quality for agricultural decision-making with emphasis to evapotranspiration calculation: A robust framework integrating dynamic time warping, fuzzy logic, and machine learning,” AgriEngineering, 2025, doi: 10.3390/agriengineering7060174.
[7.] Darban et al., “Domain adaptation contrastive learning for anomaly detection in multivariate time series,” arXiv preprint arXiv:2404.11269, 2024.
[8.] Theofrida et al., “Using wavelet transform and hybrid CNN-LSTM models on VOC and ultrasound IoT sensor data for non-visual maize disease detection,” Heliyon, 2024, doi: 10.1016/j.heliyon.2024.e26647.
[9.] Chevtchenko et al., “Anomaly detection in industrial machinery using IoT devices and machine learning: A systematic mapping,” IEEE Access, 2023, doi: 10.1109/ACCESS.2023.3333242.
[10.] Deforce et al., “Self-supervised anomaly detection of rogue soil moisture sensors,” arXiv preprint arXiv:2305.05495, 2023.
[11.] Yang et al., “Hybrid cloud-edge collaborative data anomaly detection in industrial sensor networks,” arXiv preprint arXiv:2204.09942, 2022.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26nov646

[12.] Mahmud et al., "Anomaly detection on data streams for smart agriculture," Sensors, 2023, doi: 10.3390/s23041818.
[13.] S. Shikhar and A. Sobti, "Label-free anomaly detection in aerial agricultural images with masked image modeling," arXiv preprint arXiv:2404.08931, 2024.
[14.] Gauch et al., "Rainfall-runoff prediction at multiple timescales with a single long short-term memory network," arXiv preprint arXiv:2010.07921, 2020.
[15.] Defard et al., "A patch distribution modeling framework for anomaly detection and localization," arXiv preprint arXiv:2011.08785, 2020.
[16.] Rahman et al., "Advancing cucumber disease detection in agriculture through machine vision and drone technology," arXiv preprint arXiv:2409.12350, 2024.
[17.] Khanal et al., "Paddy disease detection and classification using computer vision techniques: A mobile application to detect paddy disease," arXiv preprint arXiv:2412.05996, 2024.
[18.] Sawan et al., "Machine learning based method for forecasting crop yield," Journal of Recent Innovations in Computer Science and Technology, 2025, doi: 10.70454/JRICST.2025.20303.
[19.] Dangi et al., "A multi-temporal multi-spectral attention-augmented deep convolution neural network with contrastive learning for crop yield prediction," arXiv preprint arXiv:2509.15966, 2025.
[20.] Zulfikhar et al., "Utilization of smart agricultural technology to improve resource efficiency in agro-industry," West Science Agro, 2024, doi: 10.58812/wsa.v2i01.656.
[21.] Hasib and Akib, "An IoT-based smart plant monitoring and irrigation system with real-time environmental sensing, automated alerts, and cloud analytics," arXiv preprint arXiv:2601.15830, 2026.
[22.] Joshi et al., "Standardizing and centralizing datasets to enable efficient training of agricultural deep learning models," arXiv preprint arXiv:2208.02707, 2022.
[23.] Rahman and Azharuddin, "Wireless sensor networks in agriculture through machine learning: A survey," Computers and Electronics in Agriculture, 2022, doi: 10.1016/j.compag.2022.106928.
[24.] Rajpoot and Chandrakar, "A hybrid CNN-LSTM deep learning framework for enhanced crop yield prediction using spatial-temporal agricultural data," 2025, doi: 10.22271/maths.2025.v10.i12Sa.2200.
[25.] Rashid et al., "A comprehensive review of crop yield prediction using machine learning approaches with special emphasis on palm oil yield prediction," IEEE Access, 2021, doi: 10.1109/ACCESS.2021.3075159.
[26.] Wang et al., "LGCA-Net: A time-series anomaly detection method for multiple types of agricultural sensors," 2023, doi: 10.1016/j.inpa.2026.01.008.
[27.] Gong et al., "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," arXiv preprint arXiv:1904.02639, 2019.
[28.] Jiang et al., "SoftPatch: Unsupervised anomaly detection with noisy data," arXiv preprint arXiv:2403.14233, 2024.
[29.] Liu et al., "Precision fertilization and irrigation: Progress and applications," AgriEngineering, 2022, doi: 10.3390/agriengineering4030041.
[30.] Zhao et al., "A novel fault diagnosis method based on CNN and LSTM and its application in fault diagnosis for complex systems," Artificial Intelligence Review, 2022, doi: 10.1007/s10462-021-09993-z.
[31.] Karim et al., "A comprehensive investigation of anomaly detection methods in deep learning and machine learning," 2024, doi: 10.1049/2024/8821891.