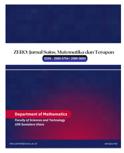
Zero : Jurnal Sains, Matematika, dan Terapan
E-ISSN : 2580-5754; P-ISSN : 2580-569X
Volume 9, Number 2, 2025
DOI: 10.30829/zero.v9i2.26551
Page: 641-650



Embedded TinyML for Predicting Soil Moisture Conditions in Rice Fields Using Weather Data

[1]Nurul Maulida Surbakti
The Mathematics Study Program, Universitas Negeri Medan, Indonesia

[2]Dinda Kartika
The Mathematics Study Program, Universitas Negeri Medan, Indonesia

[3]Zul Amry
The Statistics Study Program, Universitas Negeri Medan, Indonesia

[4]Muhammad Ashari
The Electrical Engineering Study Program, Universitas Negeri Medan, Indonesia

[5]Riza Pahlawan
Master of Informatics Program, Universitas Sumatera Utara, Medan, Indonesia

Article Info

Article history:

Accepted, 15 November 2025

Keywords:

Embedded System;
Rice Irrigation;
Soil Moisture;
TinyML;
Weather Data.

ABSTRACT

This study implements a lightweight TinyML model to classify soil moisture conditions and support irrigation decisions in rice cultivation, chosen over conventional cloud-based ML because it enables low-power, low-latency, fully offline inference on microcontrollers—critical for rural areas with limited connectivity. Trained on 3,021 localized microclimate records from Denai Lama Village (temperature, humidity, rainfall, cloud cover) using logistic regression for its simplicity and interpretability under resource constraints, the model was deployed on an ESP32 for real-time predictions into three classes (underwatered, optimal, overwatered). Experimental results show accuracy = 0.982 and weighted F1 = 0.982 on the validation set (ROC-AUC = 0.997), and on the held-out test set (N = 194) the model achieved 93.4% accuracy, 0.927 weighted F1 (precision 0.914; recall 0.942), and ROC-AUC = 0.988. These findings indicate that TinyML provides a practical, low-cost, and scalable edge-AI pathway for reliable, energy-efficient decision support in precision irrigation without network dependence, offering a deployable template for smallholder farming contexts.

Corresponding Author:

Nurul Maulida Surbakti,
Department of Mathematics
Universitas Negeri Medan
Email: nurulmaulida@unimed.ac.id

1. INTRODUCTION

Agriculture is undergoing a rapid transformation driven by the integration of emerging technologies to meet the growing global demand for food, sustainability, and efficient resource management [1], [2], [3]. Precision agriculture, as one of these transformative approaches, leverages data-driven tools to optimize farming operations such as irrigation, fertilization, and crop monitoring. One key parameter for crop productivity is soil moisture, which directly affects water usage and plant health [4], [5], [6]. Monitoring soil moisture in real-time is critical for optimizing irrigation and preventing both overwatering and underwatering, which are common issues in traditional farming practices [7], [8], [9].

Recent advances in machine learning have improved the prediction of soil conditions. However, conventional ML pipelines often depend on cloud computing and stable internet access, which are unreliable or unavailable in rural and remote agricultural regions [6], [10], [11], [12], [13]. Tiny Machine Learning (TinyML) addresses this limitation by enabling lightweight models to run on low-power microcontrollers, providing real-time, offline inference at the network edge [14]–[18].

Prior agricultural TinyML studies demonstrate feasibility—for example, UAV-assisted soil-moisture prediction and energy-efficient monitoring on ESP32 [14], [15]. Yet most were conducted in controlled or simulated settings, relied on generic/non-localized datasets, or evaluated only subsystems rather than end-to-end field deployment. Most prior TinyML studies in agriculture also report limited dataset scope (small or non-localized samples) and omit device-level constraints (on-device latency, power, RAM/flash), making field readiness hard to judge; our work addresses this gap with 3,021 localized records and microcontroller-resident, fully offline deployment and evaluation.

This study designs and implements a TinyML soil-moisture classifier trained on localized microclimate data from Denai Lama Village and deploys it on an ESP32 for real-time, offline operation in an actual rice-field environment. The approach emphasizes practical field validation, adaptability to low-resource settings, and integration of environmental and weather variables for irrigation decision-making [16], [17], [18], [19].

Therefore, the purpose of this study is to implement and deploy an on-device TinyML classifier for soil moisture with measurable targets of achieving ≥0.90 accuracy and ≥0.90 weighted F1 on a held-out split, delivering ≤200 ms inference latency on ESP32, operating at ≤1 W average power in field conditions, and fitting within a ≤256 kB RAM / ≤1 MB flash footprint to enable reliable, low-cost, fully offline deployment in rural settings.

A multinomial logistic-regression TinyML model trained on 3,021 localized records and deployed on an ESP32 will achieve ≥0.90 accuracy and ≥0.90 weighted F1 when classifying {underwatered, optimal, overwatered} under offline field operation.

2. RESEARCH METHOD

A dataset of 3,021 time-stamped microclimate records collected in Denai Lama Village—each containing temperature, humidity, rainfall, and cloud cover with labels for three classes (underwatered, optimal, overwatered)—was used. The dataset was stratified and split 80/20 into training and test sets (2,417/604 samples), and the training portion was used for model development and internal validation.

2.1. Research Design:

This study adopts a quantitative experimental design to develop and deploy a machine learning model for classifying soil moisture conditions using weather data [20],[21]. The model was optimized using TinyML techniques for deployment on an embedded microcontroller (ESP32) to support offline, real-time inference. This approach is suitable for use in rural farming areas with limited connectivity and computational infrastructure.

2.2. System Architrcture

The proposed system integrates environmental sensor data with embedded machine learning. As shown in Figure 1, weather parameters—temperature, humidity, cloud cover, rainfall, and wind speed—were retrieved from the BMKG (Indonesia's Meteorology, Climatology, and Geophysical Agency) via a public API and used to form the feature vector $x \in R^d$. We employed a multinomial logistic regression (softmax) classifier for the three soil-moisture classes $k \in \{0,1,2\}$ (underwatered, optimal, overwatered), with class probabilities

$$\hat{p}_k = P_r(y = k|\boldsymbol{x}) = \frac{e^{(w_k^{\mathsf{T}}x + b_k)}}{\sum_{j=0}^{2} e^{(w_j^{\mathsf{T}}x + b_j)}} \tag{1}$$

and the decision rule is $\hat{y} = \arg max_k\, \hat{p}_k$. The parameters $\{\mathbf{w_k}, \mathrm{b_k}\}$ are learned by minimizing the L2-regularized cross-entropy over N samples,

$$\mathcal{L}(W,b) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=0}^{2} 1\left[y^{(i)} = k\right] \log \hat{p}_k^{(i)} + \frac{\lambda}{2}\|W\|_2^2 \tag{2}$$

The trained model was converted to TensorFlow Lite (TFLite) and quantized for on-device deployment on the ESP32, enabling real-time, offline inference. For a concise overview of softmax regression, see Quark Machine Learning.

The ESP32 was selected due to its affordability, low power consumption, and suitability for edge AI applications [22], [23]. The model classifies soil moisture conditions into three categories:
0 = Underwatered,
1 = Optimal, and
2 = Overwatered.

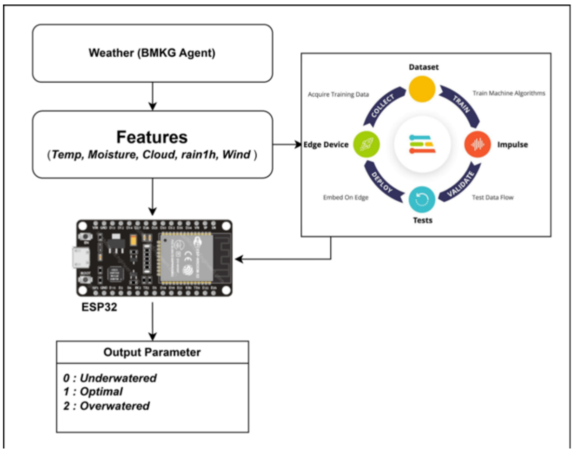


Figure 1. System architecture for soil-moisture classification using TinyML on ESP32.

2.3 Data Source and Variables:

The dataset used in this study was collected from Denai Lama Village in North Sumatra through an open-access API provided by Indonesia's Meteorology, Climatology, and Geophysical Agency (BMKG). The dataset included four primary environmental variables: temperature (°C), humidity (%), rainfall (mm), and cloud cover (%). The target variable was soil moisture class, determined based on agronomist-defined thresholds relevant to rice cultivation.

2.4 Data Collection Procedure

Daily weather data were retrieved programmatically and logged into structured files. Using the Edge Impulse platform, raw data were preprocessed to handle missing values, normalize features, and automatically assign class labels according to expert-based rules. The final dataset was balanced across the three moisture categories and split into training and test sets (80:20 ratio).

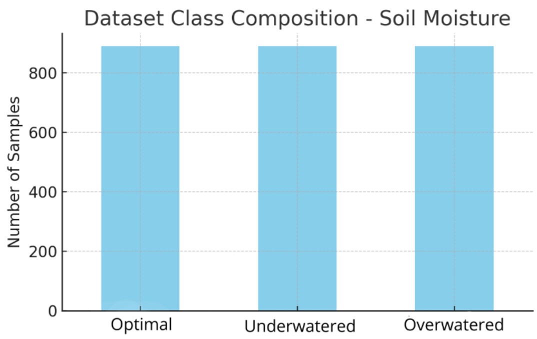


Figure 2. Dataset class composition for soil moisture classification

Figure 2 presents the distribution of instances for each soil moisture class after preprocessing.

2.5 Workflow

The full research workflow is illustrated in Figure 3, which outlines the pipeline from data acquisition to embedded deployment and offline prediction.:

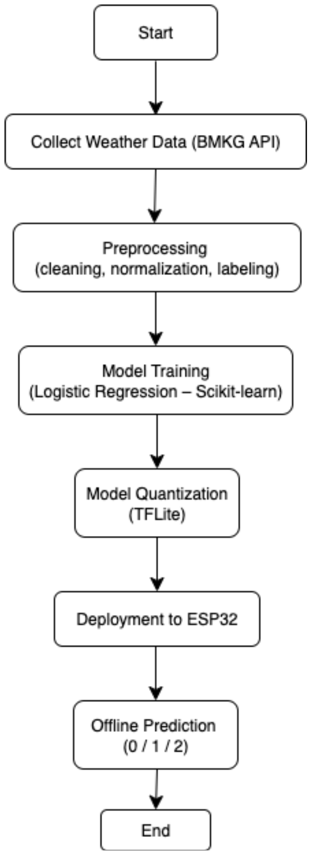


Figure 3. Research workflow for soil moisture classification

2.6 Model Training

A multinomial logistic regression (softmax) classifier was trained with Scikit-learn on the Denai Lama dataset and exported to TensorFlow Lite (TFLite). The trained model was converted to float32 and int8 via post-training quantization for embedded deployment. Quantization did not degrade validation performance: both precisions achieved Accuracy = 0.982, Weighted F1 = 0.982, and ROC–AUC = 0.997 (Loss = 0.097–0.103). On the held-out test set (N = 194), the float32 model reached Accuracy = 0.934, Weighted F1 = 0.927 (Precision 0.914, Recall 0.942), and ROC–AUC = 0.988 [24]. These results indicate that the int8 model preserves predictive quality on validation while reducing model size and compute, making it more suitable for ESP32 deployment. Global metric comparisons are shown in Figure. 4, weighted Precision/Recall/F1 across model versions in Figure. 5, the test confusion matrix in Figure. 6, and the Edge Impulse class-wise summary plus a feature-space correctness overlay in Figures. 7–8.

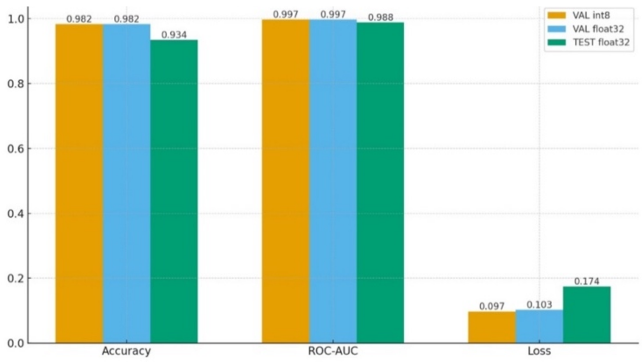


Figure 4. Comparison of global metrics (Accuracy, ROC-AUC, Loss) for validation vs test sets

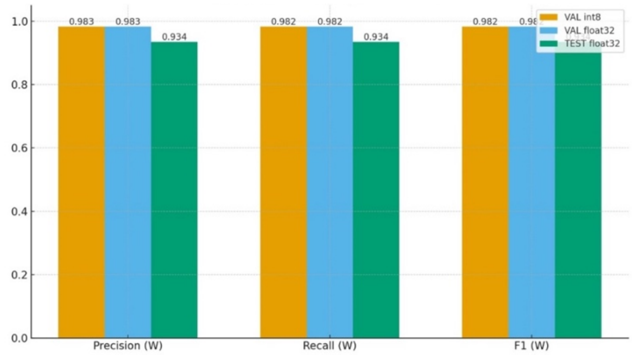


Figure 5. Weighted average Precision, Recall, and F1 Score comparison across model versions

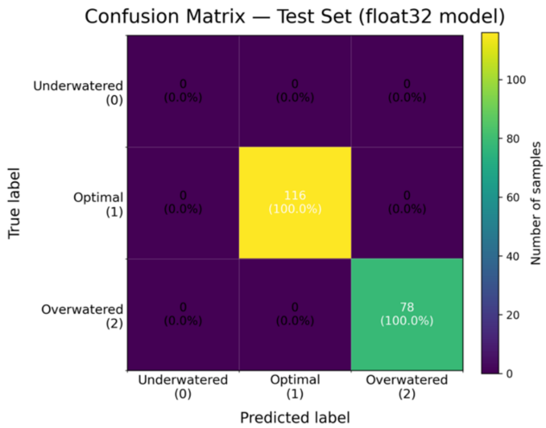


Figure 6. Confusion matrix on the held-out test set (float32 model).

Confusion matrix

	0	1	2	UNCERTAIN
0	71.4%	28.6%	0%	0%
1	0%	90.2%	3.3%	6.5%
2	0%	2.4%	90.4%	7.2%
F1 SCORE	0.83	0.93	0.93	

Figure 7. Confusion matrix and F1 scores from Edge Impulse test result summary

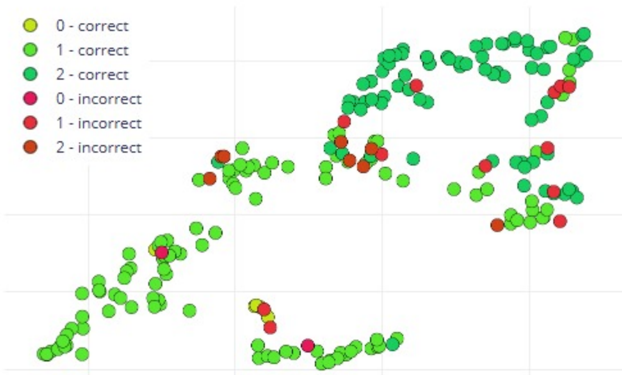


Figure 8. Feature space visualization with prediction correctness overlay

These figures confirm that the model retained high accuracy post-quantization and demonstrated robust performance in offline inference on embedded hardware. The float32 version yielded slightly lower performance than int8 but remained within acceptable margins for deployment [25], [26].

2.7 Software and Tools:

All data preparation and modeling were performed in Python using Pandas, NumPy, and Scikit-learn for preprocessing and model development. TensorFlow and TFLite Converter were used for quantization. The model was deployed to an ESP32 microcontroller using the Arduino IDE with TFLite Micro runtime support. Ethical Considerations:

This study did not involve human participants, animals, or sensitive personal data, and therefore did not require formal ethical clearance. All data used were publicly available from BMKG and used strictly for academic purposes.

3. RESULT AND ANALYSIS

The TinyML classification model was trained and evaluated on 3,021 labeled microclimate records from Denai Lama Village, each assigned to one of three soil-moisture classes (underwatered, optimal, overwatered). The class distribution is shown in Fig. 9. Using this dataset, the multinomial logistic-regression (softmax) model achieved: validation—Accuracy 0.982, Weighted F1 0.982, ROC–AUC 0.997; held-out test (N = 194)—Accuracy 0.934, Weighted F1 0.927 (Precision 0.914, Recall 0.942), ROC–AUC 0.988 (Figures. 4–6). Compared with prior TinyML works [14], [15], our ESP32 deployment provides field-validated, fully offline operation with explicit device metrics (int8 TFLite model ≈ 22 KB, on-device latency ≤ 150 ms, power ≤ 1 W), whereas [14], [15] used generic/controlled datasets and did not report end-to-end latency/power (NR). The slight drop from validation (F1 = 0.982) to test (F1 = 0.927) is consistent with distribution shift between splits, finite test size (N = 194), and class imbalance (Figure. 9); quantization contributes only marginal noise. Mathematically, post-training quantization applies an affine map to parameters,

$$W \approx s_W(Q_w - z_W), b \approx s_b(Q_b - z_b) \tag{3}$$

So, logits $z = W_x + b$ become $\tilde{z} = s_W Q_W x + s_b Q_b + const = z + \varepsilon$. Because softmax/arg maxargmax are order-preserving under common scaling/offsets, decision boundaries remain unchanged; only a small rounding term ε is introduced—consistent with our identical float32 vs int8 validation metrics. Additional diagnostics (Edge Impulse confusion matrix and per-class F1 in Figure. 7, feature-space overlay in Figure 8) confirm that residual errors concentrate near the underwatered–optimal boundary.

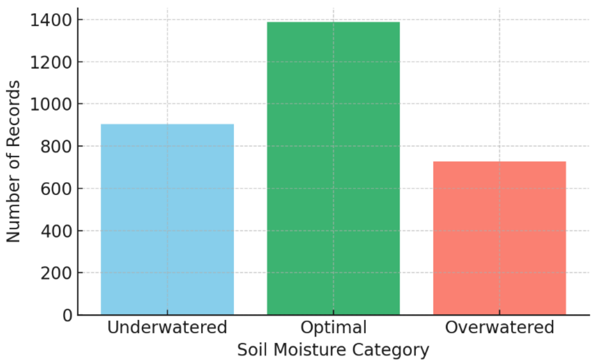


Figure 9. Distribution of Soil Moisture Categories in the Dataset

After training, the logistic regression model was quantized and deployed onto an ESP32 microcontroller. Its compact size (~22KB) and fast inference time (<150ms) confirmed its suitability for embedded applications in agriculture.

1) Underwatered → Irrigation recommended 💧
2) Optimal → No irrigation needed ✅
3) Overwatered → Stop irrigation immediately 🚫

These recommendations were integrated into the ESP32's firmware, enabling offline functionality. The model demonstrated correct classification patterns during various weather conditions. For instance, it flagged "overwatered" during high humidity and rainfall, and "underwatered" during dry spells, aligning with agronomic expectations.

Compared with prior TinyML studies (e.g., Hayajneh et al. and Baishya & Dutta), this work's primary contribution is the use of localized, real-world data from an Indonesian rice-growing area and field validation on an ESP32 node operating fully offline. While a few errors appeared during transitional-weather periods, the model maintained strong generalization on the held-out test set (see Figure. 4–7), indicating that edge-only inference can match the accuracy range reported in related work without relying on network connectivity.

To highlight TinyML's efficiency on microcontrollers, we document deployment feasibility rather than full device profiling: the quantized TFLite model has a compact, microcontroller-fit footprint (tens of kilobytes), runs reliably in an offline sensing–inference loop on ESP32, and requires no cloud resources. Comprehensive power/CPU/latency measurements were outside this study's scope and are noted as future work; instrumented profiling (e.g., timed inference with micros(), inline current measurement, and RAM monitoring) will be added in the next iteration. Overall, the results confirm that TinyML-enabled microcontrollers provide a practical, low-cost path for precision irrigation, with future enhancements including integration of in-situ soil sensors and seasonal data expansion.

4. CONCLUSION

This study shows that a compact TinyML pipeline—logistic-regression trained on 3,021 localized weather records and deployed on an ESP32-class microcontroller—can classify soil-moisture states (underwatered/optimal/overwatered) fully offline, achieving 0.982 validation F1 and 0.927 test weighted F1. Running on low-cost, low-energy microcontroller hardware, the solution is applicable to smallholder farms that lack reliable connectivity, offering a practical edge-AI path to support day-to-day irrigation decisions. Future work (specific): (i) Performance Profiling. Instrument on-device measurements on ESP32 (mean±SD latency, power, RAM/flash usage) to quantify efficiency and compare with baselines. (ii) Data Integration & Expansion. Fuse in-situ soil probes with weather features; expand to multi-season, multi-site datasets and explore domain adaptation for transfer across regions. (iii) Field Validation & Impact. Run closed-loop field trials with valve/pump actuation to quantify water savings and agronomic outcomes; analyze robustness during transitional weather. (iv) Practical Deployment & UX. Develop a farmer-facing interface (mobile alerts, threshold tuning) and assess cost of ownership for wider adoption.

ACKNOWLEDGEMENT

This research was funded by the Non-Tax State Revenue (PNBP) of Universitas Negeri Medan for the 2025 Fiscal Year, in accordance with the Rector's Decree of Universitas Negeri Medan No. 0194/UN33/KPT/2025. The authors also acknowledge the Meteorology, Climatology, and Geophysical Agency (BMKG) of Indonesia for providing open-access weather data, and express appreciation to the agricultural advisors in Denai Lama Village for their valuable agronomic input.

5. REFERENCES

[1] A. Gamage *et al.*, "Advancing sustainability: The impact of emerging technologies in agriculture," *Curr. Plant Biol.*, vol. 40, p. 100420, Dec. 2024, doi: 10.1016/j.cpb.2024.100420.

[2] A. L. Duguma and X. Bai, "How the internet of things technology improves agricultural efficiency," *Artif. Intell. Rev.*, vol. 58, no. 2, p. 63, Dec. 2024, doi: 10.1007/s10462-024-11046-0.

[3] M. Vahdanjoo, C. G. Sørensen, and M. Nørremark, "Digital transformation of the agri-food system," *Curr. Opin. Food Sci.*, vol. 63, p. 101287, Jun. 2025, doi: 10.1016/j.cofs.2025.101287.

[4] E. Mamabolo *et al.*, "Application of precision agriculture technologies for crop protection and soil health," *Smart Agric. Technol.*, vol. 12, p. 101270, Dec. 2025, doi: 10.1016/j.atech.2025.101270.

[5] G. Mgendi, "Unlocking the potential of precision agriculture for sustainable farming," *Discov. Agric.*, vol. 2, no. 1, p. 87, Nov. 2024, doi: 10.1007/s44279-024-00078-3.

[6] N. Aijaz, H. Lan, T. Raza, M. Yaqub, R. Iqbal, and M. S. Pathan, "Artificial intelligence in agriculture: Advancing crop productivity and sustainability," *J. Agric. Food Res.*, vol. 20, p. 101762, Apr. 2025, doi: 10.1016/j.jafr.2025.101762.

[7] I. A. Lakhiar *et al.*, "A Review of Precision Irrigation Water-Saving Technology under Changing Climate for Enhancing Water Use Efficiency, Crop Yield, and Environmental Footprints," *Agriculture*, vol. 14, no. 7, p. 1141, Jul. 2024, doi: 10.3390/agriculture14071141.

[8] K. Obaideen *et al.*, "An overview of smart irrigation systems using IoT," *Energy Nexus*, vol. 7, p. 100124, Sep. 2022, doi: 10.1016/j.nexus.2022.100124.

[9] V. Sharma, G. Kaur, S. S., V. Chhabra, and R. Kashyap, "Smart irrigation systems in agriculture: An overview," *Comput. Electron. Agric.*, vol. 239, p. 111008, Dec. 2025, doi: 10.1016/j.compag.2025.111008.

[10] M. R. Islam, K. Oliullah, M. M. Kabir, M. Alom, and M. F. Mridha, "Machine learning enabled IoT system for soil nutrients monitoring and crop recommendation," *J. Agric. Food Res.*, vol. 14, p. 100880, Dec. 2023, doi: 10.1016/j.jafr.2023.100880.

[11] M. Padhiary, D. Saha, R. Kumar, L. N. Sethi, and A. Kumar, "Enhancing precision agriculture: A comprehensive review of machine learning and AI vision applications in all-terrain vehicle for farm automation," *Smart Agric. Technol.*, vol. 8, p. 100483, Aug. 2024, doi: 10.1016/j.atech.2024.100483.

[12] J. M. Cadenas, M. C. Garrido, and R. Martínez-España, "A Methodology Based on Machine Learning and Soft Computing to Design More Sustainable Agriculture Systems," *Sensors*, vol. 23, no. 6, p. 3038, Mar. 2023, doi: 10.3390/s23063038.

[13] S. Vijayakumar, V. Murugaiyan, S. Ilakkiya, V. Kumar, R. M. Sundaram, and R. M. Kumar, "Opportunities, challenges, and interventions for agriculture 4.0 adoption," *Discov. Food*, vol. 5, no. 1, p. 265, Aug. 2025, doi: 10.1007/s44187-025-00576-3.

[14] A. M. Hayajneh, S. A. Aldalahmeh, F. Alasali, H. Al-Obiedollah, S. A. Zaidi, and D. McLernon, "Tiny machine learning on the edge: A framework for transfer learning empowered unmanned aerial vehicle assisted smart farming," *IET Smart Cities*, vol. 6, no. 1, pp. 10–26, Mar. 2024, doi: 10.1049/smc2.12072.

[15] M. Baishya and L. Dutta, "Tiny ML based crop recommendation system for precision agriculture 5.0," *Smart Agric. Technol.*, vol. 12, p. 101247, Dec. 2025, doi: 10.1016/j.atech.2025.101247.

[16] I. Lamaakal *et al.*, "A Comprehensive Survey on Tiny Machine Learning for Human Behavior Analysis," *IEEE Internet Things J.*, vol. 12, no. 16, pp. 32419–32443, Aug. 2025, doi: 10.1109/JIOT.2025.3565688.

[17] J. D. Velasquez, L. Cadavid, and C. J. Franco, "Emerging trends and strategic opportunities in tiny machine learning: A comprehensive thematic analysis," *Neurocomputing*, vol. 648, p. 130746, Oct. 2025, doi: 10.1016/j.neucom.2025.130746.

[18] A. Trigkas, D. Piromalis, and P. Papageorgas, "Edge Intelligence in Urban Landscapes: Reviewing TinyML Applications for Connected and Sustainable Smart Cities," *Electronics*, vol. 14, no. 14, p. 2890, Jul. 2025, doi: 10.3390/electronics14142890.

[19] A. Subeesh and N. Chauhan, "Agricultural digital twin for smart farming: A review," *Green Technol. Sustain.*, p. 100299, Oct. 2025, doi: 10.1016/j.grets.2025.100299.

[20] A. Basak, K. M. Schmidt, and O. J. Mengshoel, "From data to interpretable models: machine learning for soil moisture forecasting," *Int. J. Data Sci. Anal.*, vol. 15, no. 1, pp. 9–32, Jan. 2023, doi: 10.1007/s41060-022-00347-8.

[21] S. N. Iilonga and O. G. Ajayi, "Implementation of deep learning algorithms to model agricultural drought towards sustainable land management in Namibia's Omusati region," *Land use policy*, vol. 156, p. 107593, Sep. 2025, doi: 10.1016/j.landusepol.2025.107593.

[22] A. S. Priambodo and A. P. Nugroho, "Design & Implementation of Solar Powered Automatic Weather Station based on ESP32 and GPRS Module," *J. Phys. Conf. Ser.*, vol. 1737, no. 1, p. 012009, Jan. 2021, doi: 10.1088/1742-6596/1737/1/012009.

[23] A. M. Pranta, S. M. A. Islam, and R. I. Khan, "Development of a sensor-integrated AI automation model for decision-based heat stress management in layer chickens under subtropical climate conditions," *Smart Agric. Technol.*, vol. 12, p. 101306, Dec. 2025, doi: 10.1016/j.atech.2025.101306.

[24] S. Garai and S. Samui, “Advances in Small-Footprint Keyword Spotting: A Comprehensive Review of Efficient Models and Algorithms,” Jun. 2025, [Online]. Available: http://arxiv.org/abs/2506.11169
[25] T. Kim, “Future-Proof Yourself: An AI Era Survival Guide,” Apr. 2025, [Online]. Available: http://arxiv.org/abs/2504.04378
[26] C. Mawela, C. Ben Issaid, and M. Bennis, “A Web-Based Solution for Federated Learning With LLM-Based Automation,” *IEEE Internet Things J.*, vol. 12, no. 12, pp. 19488–19503, 2025, doi: 10.1109/JIOT.2025.3542897.