ADVANCES IN CYBER-PHYSICAL SYSTEMS
Vol. 10, No. 2, 2025

SMART PLANT WATERING USING TINYML: WATER SAVINGS THROUGH PREDICTIVE CONTROL

Roman Korostenskyi, Igor Olenych

Ivan Franko National University of Lviv, 50, Drahomanova str., Lviv, 79005, Ukraine.
Authors' e-mails: *roman.korostenskyi@lnu.edu.ua, igor.olenych@lnu.edu.ua*

https://doi.org/10.23939/acps2025.02.141

Submitted on 02.10.2025

***Abstract*: Indoor plant watering is not always effective - people often overwater or underwater plants, wasting water and harming plant health. In view of this, a smart watering system using artificial intelligence that runs on a tiny microcontroller chip has been developed. The proposed system predicts when plants need water and waters them automatically. Testing on 12 plants for 3 months has showed 27 % water savings versus manual watering and 15 % savings versus simple automated systems. The AI model is only 8.7 KB and runs for months on battery power without Internet. This proves that tiny AI can save water and improve plant care.**

***Index terms*: Smart irrigation, TinyML, Edge AI, water conservation, IoT, predictive control.**

I. INTRODUCTION

Indoor plant cultivation is a massive global activity, with millions of households and commercial spaces maintaining plants. However, conventional watering practices are often inefficient. Studies show that 30-40 % of water used for indoor plants is wasted through overwatering or runoff [1].

The fundamental problem is that most people water plants on fixed schedules (e.g., every Monday morning with a fixed volume) regardless of actual plant needs. This approach fails because plant water consumption varies dramatically based on:

- Environmental conditions (temperature, humidity, light intensity).
- Plant growth stage and health status.
- Seasonal variations in metabolism.
- Soil moisture retention characteristics.

Fixed schedules lead to overwatering during cool periods (causing root rot and runoff) and underwatering during warm periods (causing drought stress). This not only wastes water but also harms plant health, contributing to the perception that plant care is difficult.

Several "smart" irrigation solutions exist, but each has significant drawbacks.

Cloud-dependent systems require constant Internet connectivity and transmit all sensor data to remote servers for processing. This creates reliability issues (failure during internet outages), privacy concerns (home environmental data transmitted externally), and increases energy consumption from continuous wireless communication.

Simple threshold-based systems use basic reactive control: they water when soil moisture falls below a threshold. While better than fixed schedules, this reactive approach allows plants to experience stress before watering occurs. The control law is simply

$$Water(t) = \begin{cases} 1 & \text{if } m_t < \theta \\ 0 & \text{otherwise} \end{cases}, \tag{1}$$

where m_t is the current moisture, and θ is the threshold value. Such systems can operate autonomously and consume less energy than wireless irrigation systems.

It should be noted that battery life limitations plague most wireless systems because they continuously transmit data or maintain active processors, requiring frequent battery replacement.

II. RELATED WORK AND BACKGROUND

A. IOT SYSTEM VALIDATION AND RELIABILITY

Tritchkov [2] conducted a systematic mapping study on verification and validation methodologies for industrial third-party IoT applications, identifying key challenges in ensuring system reliability in Industry 4.0 contexts. These findings are particularly relevant for agricultural IoT deployments where system failures can result in crop loss. Our edge-based approach addresses several validation concerns by maintaining local operation capabilities even during network failures.

B. SMART IRRIGATION SYSTEMS

Machine learning-based irrigation has been extensively studied for both agricultural and indoor applications. Early IoT-based systems focused on basic sensor networks with threshold controls [3], achieving modest water savings (10-15%) but suffering from reactive behavior.

More recent research has integrated machine learning for improved decision-making. Kashyap et al. [4] demonstrated deep learning models achieving high irrigation accuracy, but their system required cloud computing infrastructure. A comprehensive review [5] found that ML-based irrigation systems typically report 20-30% water savings compared to manual methods, but most implementations depend on powerful computing resources.

The fundamental challenge is balancing computational sophistication with practical deployment constraints. Cloud-based systems offer unlimited computing but suffer from latency, reliability, and privacy issues. Local processing offers autonomy but requires careful resource management.

C. EDGE COMPUTING FOR AGRICULTURE

Edge computing - processing data locally rather than in the cloud - has emerged as a compromise solution. Premkumar and Sigappi [6] showed that edge-based irrigation reduces latency and improves reliability compared to cloud systems. However, their work used Raspberry Pi devices (quad-core processors, 1+ GB RAM, 5+ watt power consumption), which are impractical for battery-powered plant nodes.

The gap between cloud computing and edge servers leaves an important niche: ultra-low-power microcontrollers that can run for months on small batteries while still executing AI models.

D. TINYML: MACHINE LEARNING ON MICROCONTROLLERS

Recent advances in TinyML (Tiny Machine Learning) enable neural networks to run on microcontrollers with only kilobytes of memory [7]. Key enabling technologies include:

*Model Quantization***.** Converting neural network weights from 32-bit floating point to 8-bit integers can reduce model size by 4-10× with minimal accuracy loss [8]. The quantization process maps floating-point values to integers:

$$q(t) = round\left(\frac{r-z}{s}\right), \tag{2}$$

where r is the real (float) value, q is the quantized (int8) value, S is the scale factor, and z is the zero-point offset.

Model Pruning. Removing unnecessary neural network connections can reduce inference time and memory footprint while maintaining accuracy.

Knowledge Distillation. Training smaller "student" networks to mimic larger "teacher" networks allows for the compression of complex models.

While TinyML has been demonstrated for various applications [9,10], no prior work has shown practical multi-month deployment for autonomous plant care. Our work fills this gap by demonstrating that TinyML can long-term power real-world IoT applications.

III. SCOPE OF WORK AND OBJECTIVES

We proposed a different approach to indoor plant watering using Edge AI – artificial intelligence that runs directly on the microcontroller attached to each plant. This enables predictive rather than reactive irrigation control.

Our system learns each plant's unique moisture decline patterns and predicts future soil moisture levels. Instead of waiting for the soil to become dry, it waters proactively when it predicts moisture will soon fall below the optimal threshold. The predictive control law is

$$Water(t) = \begin{cases} 1 & \text{if } \hat{m}_{t+\Delta t} < \theta \\ 0 & \text{otherwise} \end{cases}, \tag{3}$$

where $\hat{m}_{t+\Delta t}$ is the predicted moisture at $t + \Delta t$ time.

The primary function of artificial intelligence in this study is the regulation of irrigation water quantity. The algorithm is designed to determine the required water volume necessary for maintaining soil moisture levels at 50%, utilizing multiple input parameters: historical soil moisture records (3 preceding measurements), ambient temperature and humidity. This methodology simultaneously addresses multiple challenges. Specifically, the approach enables efficient water utilization by preventing excessive water application during irrigation cycles. Furthermore, it contributes to improved plant health outcomes by maintaining soil moisture within optimal ranges, thereby precluding both severe soil desiccation and excessive saturation.

The entire AI model is compressed to just 8.7 KB through quantization techniques, allowing it to run on a microcontroller for months on battery power without Internet connectivity. This study is focused on a systematic assessment of key performance metrics of the TinyML-based smart watering system.

IV. SMART WATERING SYSTEM DESIGN

A. HARDWARE ARCHITECTURE

The indoor plant watering system was implemented according to the IoT paradigm. In particular, each plant was equipped with a microcontroller-based node. A capacitive sensor DFRobot SEN0193 was used to control soil moisture. Current values of the temperature and relative humidity of the air were measured using a DHT22 sensor. The ESP32-S3 microcontroller (8 MB Flash, 512 KB RAM) was used as a platform for collecting, processing, and analyzing sensor data and controlling the mini water pump (12V, 2.8-3.0 L/min). Additionally, a microSD card for logging and a 1200 mAh Li-ion battery were used.

B. AI MODEL DESIGN

The neural network predicts soil moisture 3 hours ahead to enable proactive watering. We chose a 3-hour prediction horizon as a balance: long enough to provide meaningful advance notice, short enough to maintain high accuracy.

Architecture. We use a feedforward neural network with the following structure:

- Input layer: 4 neurons (current moisture + 3 previous hourly readings)
- Hidden layer 1: 16 neurons with ReLU activation
- Hidden layer 2: 16 neurons with ReLU activation
- Output layer: 1 neuron (predicted moisture)

The forward pass computation is

$$h_1 = \mathrm{ReLU}(W_1\mathrm{x}+\mathrm{b}_1), \tag{4}$$

$$h_2 = \mathrm{ReLU}(W_2\mathrm{h}_1+\mathrm{b}_2), \tag{5}$$

$$\hat{m}_{t+3} = W_3\mathrm{h}_2+\mathrm{b}_3, \tag{6}$$

where $x = [m_t, m_{t-1}, m_{t-2}, m_{t-3}]^T$ is the input vector, W_i are weight matrices, b_i are bias vectors, and $\mathrm{ReLU}(z)=\max(0,z)$.

Model equation:

$$\hat{m}_{t+3} = f_{\theta}(m_t, m_{t-1}, m_{t-2}, m_{t-3}), \quad (7)$$

where $\hat{m}_{t+3}$ is predicted moisture, m_t is current moisture, and θ is the neural network with parameters $\theta = \{W_1, b_1, W_2, b_2, W_3, b_3\}$.

Training Process. We trained the model using mean squared error loss

$$L(\theta) = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{m}_i - m_i\right)^2, \quad (8)$$

with Adam optimizer (learning rate α = 0.001). Training data consisted of 1000 samples from soil dry-down curves collected over 4 weeks for a Monstera plant. We used 80/20 train/validation split.

Quantization. INT8 quantization converts 32-bit floats to 8-bit integers:

$$w_{\text{int}8} = clip\left(round\left(\frac{w_{fp32}}{s} \right) + z, -128, 127 \right), \quad (9)$$

where w is weight, S is scale factor (computed per-layer as $s = \frac{\max(|w|)}{127}$), z is zero-point offset. Quantization reduced the model size from 35 KB to 8.7 KB (4× compression) with only 0.3 % accuracy degradation.

Decision logic. The system waters when

$$\hat{m}_{t+3} < \theta_{\min} \text{ where } \theta_{\min} = 50\%. \quad (10)$$

As a result, the final model performance demonstrates 2.1 % RMSE on the validation set and 5.6 ms inference time on ESP32-S3.

V. EXPERIMENT

We conducted a controlled 90-day experiment to compare the effectiveness of the proposed smart watering system and traditional approaches. In particular, the ability to maintain stable soil moisture, the saved water volume, the accuracy of on-device moisture prediction in real deployment, and the duration of reliable battery operation of the TinyML model were analyzed to assess the effectiveness. We used 12 Monstera deliciosa (Swiss cheese plant) specimens, chosen for their popularity as houseplants, moderate water requirements, and clear visual stress signals. All plants were healthy at the experiment start of a similar size in identical pots with the same soil. The temperature was maintained at 20–24°C and the humidity at 40–60 % in a room with natural lighting. No fertilizer was applied to isolate watering effects.

The plants were divided into three groups (n=4 each) to compare irrigation strategies.

Group M (Manual Schedule) simulates typical houseplant care. Plants were watered manually every Monday morning with 250 mL of water (enough to fully saturate the pot). This volume is based on standard Monstera care recommendations. Sensor nodes were installed for data collection, but the pumps were disabled.

Group T (Threshold Automation) is equipped with conventional sensor-based automation. ESP32 node checks soil moisture hourly and activates the pump for 5 seconds (delivering ~250 mL) whenever moisture falls below 50 %.

Group E (Edge AI Predictive) is embedded with the complete smart system. AI model predicts moisture 1 hour ahead and waters proactively when the forecast drops below 50%. Hardware is identical to Group T, differing only in the control algorithm.

We defined quantitative Key Performance Indicators (KPIs): Water Efficiency, Moisture Stability, AI Prediction Error, and Plant Growth.

Water Efficiency reflects the percentage of water saved relative to manual watering

$$W_{saved} = \frac{W_{manual} - W_{system}}{W_{manual}} \times 100\%. \quad (11)$$

Moisture Stability is related to the standard deviation of soil moisture readings

$$\sigma_m = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(m_i - \bar{m}\right)^2}, \quad (12)$$

where N is the number of measurements, m_i is an individual reading, $\bar{m}$ is the mean moisture.

AI Prediction Error is determined by the root mean square error between the predicted and actual moisture

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{m}_i - m_i\right)^2}. \quad (13)$$

Plant Growth is associated with the number of new leaves and the leaf area change estimated over 90 days.

Each microcontroller-based node recorded hourly measurements of the soil moisture percentage, temperature and relative humidity of the air, irrigation events (timestamp, duration), battery voltage, and AI prediction accuracy (Group E only).

We used one-way ANOVA to test for significant differences between groups, with individual plants as replicates (n=4 per group). Significance level set at p<0.05. For ANOVA

$$F = \frac{MS_{between}}{MS_{within}} = \frac{\sum n_i \left(\bar{x}_i - \bar{x}\right)^2 / (k-1)}{\sum\sum (x_{ij} - \bar{x}_i)^2 / (N-k)}, \quad (14)$$

where k = 3 groups, N = 12 total plants.

VI. RESULTS AND DISCUSSIONS

Our Edge AI system demonstrated three major advantages over conventional approaches.

1. Water Conservation Through Predictive Timing. Table 1 shows the average value of water consumption by a plant from each group over 90 days. The 27% water savings vs manual watering and 15% savings vs threshold automation demonstrate the value of prediction, which puts this work at the high end of reported ML irrigation results [5]. The AI waters just before plants need it, avoiding both the wasteful fixed-schedule approach and the reactive threshold approach, where plants partially dry out before watering triggers.

Table 1

Water usage comparison

Group	Water/Plant	Water savings E vs M	Water savings E vs T
Group M	3.0 L	-	-
Group T	2.6 L	13 %	-
Group E	2.2 L	27 %	15%

Statistical significance ANOVA is $p < 0.01$.

The water savings can be modeled as

$$\Delta W = W_{base} - W_{optimal} = \int_{t_0}^{t_f} \left(r_{base}(t) - r_{optimal}(t) \right) dt, \quad (15)$$

where $r(t)$ is the watering rate over time. Fixed schedules have a high r_{base} regardless of need. The predictive system optimizes $r_{optimal}$ to match actual evapotranspiration.

2. Plant Health Through Moisture Stability. The results of the study on maintaining soil moisture stability are presented in Table 2. As a result of applying a predictive AI model, the irrigation system provided the greatest soil moisture stability and the smallest percentage of time when the moisture was below the threshold value. The worst results were demonstrated by manual watering of houseplants. The reduced moisture variability directly improved the growth of the plants. Leaf area increases of 9, 12, and 18% over 90 days were observed for monsters in groups M, T, and E, respectively. This aligns with plant physiology – stable soil moisture enables consistent and photosynthetic rates.

Table 2

Maintaining soil moisture stability

Group	Soil moisture variability	Time below 30% of the moisture threshold
Group M	12.5 %	29 %
Group T	9.8 %	8 %
Group E	5.2 %	1.5 %

The key difference between reactive vs predictive soil moisture control: reactive control allows soil to reach the threshold before responding, while predictive control acts in advance. This reduces peak-to-trough soil moisture swings, preventing plant stress while using less water. In particular, 15% additional water savings of predictive AI over threshold automation reveal the fundamental value of forecasting.

It is also worth noting that the duration during which the soil moisture level remained below 50% was the lowest in Group E, accounting for only 1.5% of the total experimental period. In contrast, Group M exhibited soil moisture levels below this threshold for approximately 30% of the total experiment time, indicating a substantially less stable moisture retention performance.

It is worth mentioning that the performance of the AI model after making more than 3000 forecasts according to the metrics of MAE, RMSE, and accuracy within ±5 % was 2.2, 2.7, and 92 %, respectively. Inference time was 5.6 ms per prediction.

VII. CONCLUSION

This work proves that AI can improve plant care while running on microcontrollers. The proposed smart watering system achieved:

- 27% water savings vs manual watering;
- 18% plant growth improvement;
- 15+ month battery life with 8.7 KB AI model;
- full autonomy without internet connectivity.

The predictive approach (water before plants need it) beat reactive control (water when plants are dry) by 15%. At scale, this technology could save millions of liters annually.

Edge AI eliminates cloud dependencies while enabling sophisticated decisions on tiny devices. This demonstrates that practical AI doesn't always need large models or powerful computers - sometimes the best solution runs on a low-cost chip.

VIII. CONFLICTS OF INTEREST

The authors declare no conflicts of interest.

IX. DECLARATION ON GENERATIVE AI

During the preparation of this work, the author(s) used ChatGPT, Grammarly in order to: Grammar and spelling check, Paraphrase and reword. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] Abioye, E. A., Hensel, O., Esau, T. J., Elijah, O., Abidin, M. S. Z., Ayobami, A. S., ... & Nasirahmadi, A. (2022). Precision irrigation management using machine learning and digital farming solutions. *AgriEngineering*, *4*(1), 70-103. DOI:https://doi.org/10.3390/agriengineering4010006

[2] Tritchkov, I. (2020). Systematic Mapping Study on Verification and Validation of Industrial Third-Party ioT Applications. *Advances in cyber-physical systems*, (5, Num. 1), 29-43. DOI: 10.23939/acps2020.01.030

[3] Goap, A., Sharma, D., Shukla, A. K., & Krishna, C. R. (2018). An IoT based smart irrigation management system using Machine learning and open source technologies. *Computers and electronics in agriculture*, *155*, 41-49. DOI: https://doi.org/10.1016/j.compag.2018.09.040

[4] Kashyap, P. K., Kumar, S., Jaiswal, A., Prasad, M., & Gandomi, A. H. (2021). Towards precision agriculture: IoT-enabled intelligent irrigation systems using deep learning neural network. *IEEE Sensors Journal*, *21*(16), 17479-17491. DOI: 10.1109/JSEN.2021.3069266

[5] Touil, S., Richa, A., Fizir, M., Argente Garcia, J. E., & Skarmeta Gomez, A. F. (2022). A review on smart irrigation management strategies and their effect on water savings and crop yield. *Irrigation and Drainage*, *71*(5), 1396-1416. DOI: https://doi.org/10.1002/ird.2735

[6] Premkumar, S., & Sigappi, A. N. (2022). IoT-enabled edge computing model for smart irrigation system. *Journal of Intelligent Systems*, *31*(1), 632-650. DOI: https://doi.org/10.1515/jisys-2022-0046.

[7] Tsoukas, V., Gkogkidis, A., & Kakarountas, A. (2022, November). A tinyml-based system for smart agriculture. In *Proceedings of the 26th Pan-Hellenic Conference on Informatics* (pp. 207-212). DOI: https://doi.org/10.1145/3575879.3575994.

[8] Doshi, S., Vora, K., Mashru, D. (2022). Edge AI for Low-Power IoT Devices: Architectures, Algorithms, and Applications, *Educational Administration: Theory and Practice, 28(2)*, 309-312. DOI: https://doi.org/10.53555/kuey.v28i02.10499.

[9] Elhanashi, A., Dini, P., Saponara, S., & Zheng, Q. (2024). Advancements in TinyML: Applications, limitations, and impact on IoT devices. *Electronics*, *13*(17), 3562. DOI: https://doi.org/10.3390/electronics13173562

[10] Osadchuk, O., Olenych, I. (2025). IoT System for Real-time Audio Information Processing. *Advances in Cyber-Physical Systems*, *10(1).* 16-21. DOI: 10.23939/acps2025.01.016.

Roman Korostenskyi received the Master's degree in Computer Science from Ivan Franko Lviv National University, Ukraine in 2022. He is presently a PhD student at the Department of Radioelectronic and Computer Systems at Ivan Franko Lviv National University. The main research topics are machine learning applications in embedded systems..

Igor Olenych received the degree of PhD Candidate in Physical and Mathematical Sciences from the Institute for Physical Optics, Ukraine in 2010 and the degree of Doctor of Sciences in Physics and Mathematics from Ivan Franko Lviv National University, Ukraine in 2020. He is presently the Head of the Department of Radioelectronic and Computer Systems at Ivan Franko Lviv National University. His current research interests include fuzzy modeling as well as smart solutions and systems.