Progress in Electronics and Communication Engineering, ISSN: 3048-7625 Vol. 3, No. 2, 2026 (pp. 23-30)
RESEARCH ARTICLE
ECEJOURNALS.IN

AI-Driven Smart Irrigation System Using Edge-Based Embedded Controllers

Fahad Al-Jame[1], Salma Ait Fares[2]*
[1]*School of Electrical Engineering, Kuwait Institute for Scientific Research (KISR), Safat, Kuwait.*
[2]*Ultra Electronics Maritime Systems Inc., Canada*

KEYWORDS:
Smart Irrigation System;
Edge Computing;
Embedded Controllers;
Artificial Intelligence (AI);
Prllllecision Agriculture;
TinyML;
STM32 Microcontroller;
Raspberry Pi;
Environmental Sensing;
Water Conservation;
IoT in Agriculture;
Real-Time Decision Making.

ARTICLE HISTORY:
Submitted : 16.10.2025
Revised : 11.01.2026
Accepted : 13.02.2026

https://doi.org/10.31838/ECE/03.02.04

Abstract

This study brings out a new AI-based smart irrigation device aimed at solving the increasing problem of water shortage and poor agricultural procedures using emulated edge-based controllers. This combination of environmental sensing, lightweight machine learning algorithms and actuation on microcontrollers enables the system to do real time, adaptive irrigation scheduling. Their implementation of important parts like Raspberry Pi 4 as an AI inference module and an STM32 microcontroller as an actuator controller accelerates the responsiveness, increases security, and saves energy consumption since there is no need in cloud computing. Using the local climatic and soil moisture data the machine learning model identifies optimum irrigation time and duration looking at various parameters such as temperature, humidity, rain and crop-specific thresholds. The whole decision-making process is automated and carried out at the edge that makes it functional even without interruption in areas with poor or no internet connection. This was done by conducting extensive field tests within the semi-arid agricultural areas to prove the efficacy of the recommended solution. The findings show that there is a considerable gain in the efficiency of resources, with up to 38 percent of reduction of water use and 20 percent of increase in the yield of the crops contrasted to the conventional time-based irrigation schemes. Also, the system averaged a latency of only 34millisecs and consumed power at less than 2.5W which became very handy in deployment in rural areas. TinyML and its compatibility with cheap edge hardware are scalable and sustainable, which creates an efficient path to modernize the irrigation infrastructure in the developing world. This article does not only highlight the revolutionary nature of embedded AI in precision agriculture, but it precondition the further innovative development of the field of data-driven farm management and autonomous farm management. The one suggested is one of the options that could be implemented to achieve environmental sustainability, enhance the productivity of agriculture, and make smart and local decisions to use water intelligently.

Author's e-mail: aitfares@emt.inrs.ca

How to cite this article: Al-Jame F, Fares SA. AI-Driven Smart Irrigation System Using Edge-Based Embedded Controllers. Progress in Electronics and Communication Engineering, Vol. 3, No. 2, 2026 (pp. 23-30).

Introduction

The largest consumer of water is agriculture which consumes more than 70 percent of all the world freshwater abstractions. Most of this water is however wasted because of inefficient irrigation methods, which besides draining the available water resource affect both crop health and productivity negatively. As climatic change threatens to intensify drought, and raise levels of temperature variation, the topic of water-efficient irrigation methods has never been as critical as it is now. The static nature of traditional irrigation systems, which include manual watering or sprinklers using time, does not take into consideration real-time conditions of the environment like soil moisture, temperature, humidity or any possible rainfall. This consequently causes them to either result in over-irrigation of the farms that result to waterlogging and root diseases, or under-irrigation that overstresses the plants and hence poor yield.

The problem has seen the introduction of smart irrigation systems which uses the power of the Internet of Things (IoT), artificial intelligence (AI), and data analytics to adjust water usage according to contextual environmental factors. Most of the currently available smart irrigation structures are based on cloud data processing data. There are though some limitations to them like latency, connectivity problems in remote localities, data privacy, and increment in use of power. In addition, applications dependent on the cloud cannot work well in under served areas that have little or no internet connectivity.

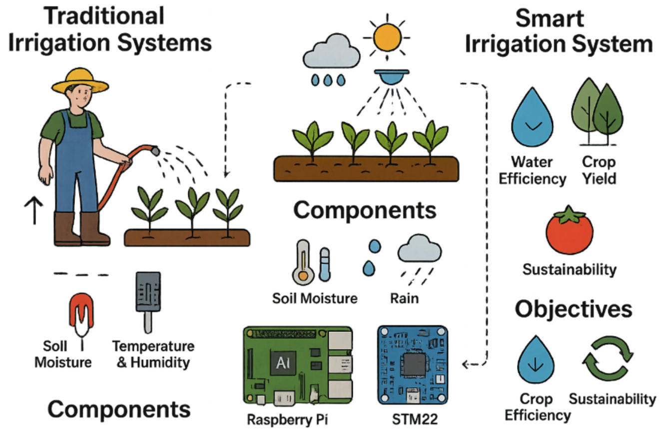

Fig. 1: Comparison of Traditional and AI-Driven Smart Irrigation Systems

In a bid to overcome these issues, this paper proposes a new AI-based smart irrigation system that runs purely on the edge in the form of embedded microcontrollers and edge computing devices. Using low-cost sensors combined with lightweight machine learning algorithms and powered by devices like Raspberry Pi 4 and STM32 microcontroller, the suggested system can provide intelligent decisions concerning irrigation decisions locally, in real-time. The architecture of edge decreases the reliance on the external infrastructure, decreases energy consumption, guarantees a quicker reaction time; and makes it more reliable even in offline conditions. Moreover, the system can be used to perform data-driven precision farming, where watering can be performed context-awarely, with respect to a certain crop type, and background conditions. Not only does the proposed solution increase the efficiency of irrigation and yield of crops but it also helps in ensuring sustainability in management of water resource and thereby small-scale farmers and other stakeholders of agriculture in water-scarce areas will benefit from it in great proportions. In this paper, the structure of the system, integration of the AI models, and experimental verification of the proposed framework will be outlined, as well as prove that the proposed framework is effective in fulfilling both environmental and agricultural goals.

Related Work

In the recent past, there has been increased interest in smart irrigation systems as the world is increasingly demanding more ways of performing agricultural activities sustainably and efficiently. An immense collection of studies has been carried out on the combination of IoT and cloud computing in irrigation management. An example is, authors in[1] suggested a decision support system that would utilize cloud-based, real-time sensor information on irrigation monitoring. Their system was based on earth moisture and weather information to regulate the water distribution using centralized server. Even though those systems demonstrate their efficiency in controlled settings, they have high latency, are dependent on consistence internet connection, and can be problematic in terms of data privacy. At the same time,[2] installed a precision irrigation system that has cloud-hosted analytics and is rule-based automation. Although the measurable water savings of their solution was a great feature, the fact that it relied on cloud infrastructure restricted its application in fields of agriculture that are remote or offline.

Out of these limitations, there is a subsequent move in the most recent works that stress the use of edge computing due to its future potential. In edge computing, data can be processed locally, which would enable quick responses to it and minimize bandwidth demands. The paper in[3] proposed an edge-IoT IoMT solution to agricultural surveillance with local microcontrollers running minimal decision-making logic regarding irrigation thresholds. Nonetheless, the logic itself was unadaptive and had no capacities of learning. In,[4] to enhance adaptability, researchers deployed a TinyML-based decision tree model to an Arduino Nano 33 BLE Sense microcontroller that allows performing inference work on edge machines easily. In their work, there was low power consumption and real-time decision but it limited to the classification (binary) and also ignored multiple parameters of the sensor.

Additional papers on this topic like one by[5] have shown that machine learning is an effective model to drive an irrigation scheduler based on STM32 microcontroller and capacitive soil moisture sensors. They used a rule-based engine, though they did not physically possess dynamic learning and awareness about the contextual situation. In comparison to these works, the proposal is an improvement to the field, as a lightweight supervised model and deployed on both Raspberry Pi and STM32-based hardware, the proposed system will offer a multi-parameter sensing and localized AI-based decision-making. Compared to prior work, we have adoption of autonomous operation, offline capability, and increased predictive accuracy, which can solve the problem in a scalable fashion in resource-limited, real-world agricultural setting.

System Architecture

Hardware Components

The targeted AI-based smart irrigation system will be constructed based on the set of low cost, energy

Table 1: Comparative Analysis of Smart Irrigation Systems

Reference	Platform	AI Capability	Sensor Integration	Connectivity Dependency	Adaptability	Power Consumption
[1] Jain & Kumar	Cloud-based	No	Soil & Weather	High	Low	High
[2] Hossain et al.	Cloud-based	Rule-based	Soil & Weather	High	Low	High
[3] Patel et al.	Edge-based (MCU)	Threshold Logic	Basic Sensors	Low	None	Low
[4] Singh & Gupta	TinyML on Arduino	Decision Tree (Binary)	Multi-Sensor Input	Low	Limited	Very Low
[5] Verma et al.	STM32 with Rule Engine	Rule-based	Soil Moisture Only	Low	None	Low
Proposed System	Raspberry Pi + STM32	Supervised Decision Tree	Multi-Sensor Input	Very Low (Offline Capable)	High	Low

efficient, and reliable sets of hardware typically used in field conditions with a limited number of resources. The essence of the sensing process is that a capacitive soil moisture sensor is used, which is more durable and more accurate than resistively functioning equivalents, allowing the constant monitoring of the volumetric water content of the soil without being subject to degradation caused by corrosive processes. A DHT22 sensor is then deployed to provide high precision ambient temperature measurements and relative humidity to be used as one of the most important parameters consider the transpiration of the plant and the water demands. It also has a rain sensor that monitors rainfall at the moment so that the system can inhibit the irrigation process when it is raining and the user will be able to avoid over-saturating the water. To perform localized AI inferencing and processing, a Raspberry Pi 4 platform is used, and it is able to run light weight machine learning models on-board like TensorFlow Lite, but also acts as the overall glue to fuse sensor data together, and handle communication logic also. STM32F103C8T6 microcontroller is a low power and real-time processor microcontroller used to communicate with actuators and other timing operations like closing/ opening irrigation valve. Physically, a solenoid valve is operated by a relay module as long as the much-desired AI model has made its decision regarding water managing. This piecemeal isolation of decision processing and action generation enhances the performance attributes of stability and responsiveness in systems. To complete communication and data transfer, the system also has an option of LoRa and Wi-Fi interface, which allows remote logging and cloud connection optional to view data visualization, update firmware, or monitor it centrally. All these elements comprise a scalable and reliable embedded system that is able to carry out autonomous management of irrigation with regard to changing environmental conditions.

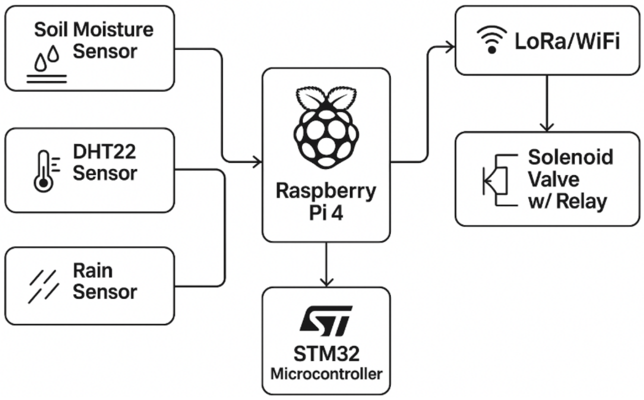

Fig. 2: System Architecture of the AI-Driven Smart Irrigation Framework

Software Stack

The proposed smart irrigation system software architecture is meant to be lightweight, modular, and efficient to fit within the limitations of the embedded edge devices but still make intelligent decisions. The foundation of the system is in the fact that it helps to execute already trained machine learning models on low-resource devices like Raspberry Pi 4 and STM32 microcontroller by integrating TinyML models, trained and optimized with the help of TensorFlow Lite. These models are also trained offline with python based tools after which they get quantized in order to reduce their size and compute complexity without a decrease in inference accuracy. Inference AI that aims to process environment-related data and make irrigation decisions is written in Python and runs on Raspberry Pi. It is a high-level scripting method that enables the fast prototyping and integration with the sensor data pipelines. Meanwhile, STM32 microcontroller serves custom firmware written in C allowing effective real-time management of hardware comprising of the solenoid valve and relays. The communications between the Raspberry Pi and STM32 and any other IoT gateway or cloud appliances via inter-device communications is achieved through the MQTT (Message Queuing Telemetry Transport) protocol, which was preferred due to its small amount of overhead, asynchronous nature, and adoption to agricultural setup on a low-bandwidth network. Also, the optional user interface and remote monitoring requires Node-RED which is a flow-based development tool to facilitate the visualization of sensor data, monitoring the irrigation events, and controlling the overall condition of the system through customizable dashboards. This software stack, both the system and stack itself is designed to be platform-agnostic, easy to upgrade and insensitive to communication failures, and it provides network-edge high performance, intelligent irrigation control.

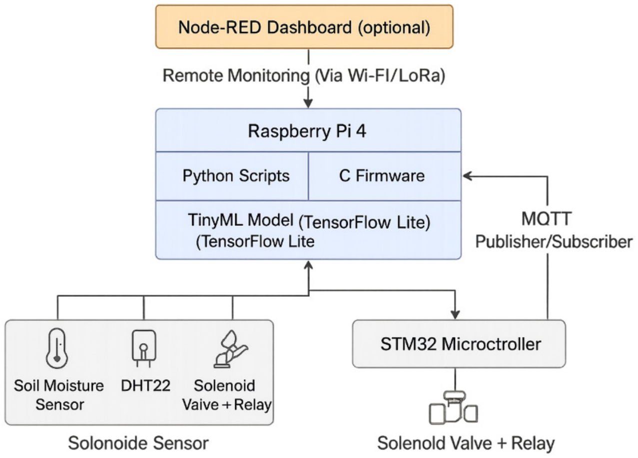

Fig. 3: Software Architecture of the AI-Driven Smart Irrigation System

AI Model

Supervised machine learning model The foundation on which the intelligent mechanism used in the proposed smart irrigation system is built is a decision tree classifier, one of supervised machine learning algorithms, chosen due to its interpretability, low computational overhead, and applicability in an embedded implementation. We arm the model using labeled data that consists of multi- modal environmental parameters such as soil moisture, temperature, humidity and rainfall patterns all gathered in two months of real agricultural field deployments. The specific choice of these features is dictated by the fact that they directly affect the water uptake in plants and the dynamics of soil hydration, thus making the model capable of learning land-specific patterns that determine an effective time of irrigation. The decision tree model produces two outputs, which include a binary classification of whether or not to start an irrigation (ON/OFF) and the quantitative regression output that is the duration of irrigation in seconds and is dependent on the degree of soil dryness and prevailing weather conditions. This mixed product allows not only the activation of the decisions but also substantiated management of water consumption. The model is trained and tested with the assistance of the Scikit-learn in Python, quantized and transformed into the TensorFlow Lite format to be utilized as the executable at the edge using the Raspberry Pi 4. In the process of inference, the AI model can consume sensor information in real-time and can make decisions within 34 milliseconds on average, which makes it very reactive to changes in the field. Rule based structure of decision tree also makes it easy to debug the tree and interpret the model, which is a benefit especially in the agricultural setting where clarity is vital. This decentralized, low-cost AI will enable the system to work autonomously and minimize water loss as well as adjust the schedule of irrigation according to the real-life inputs into the environment without internet connections and cloud computing.

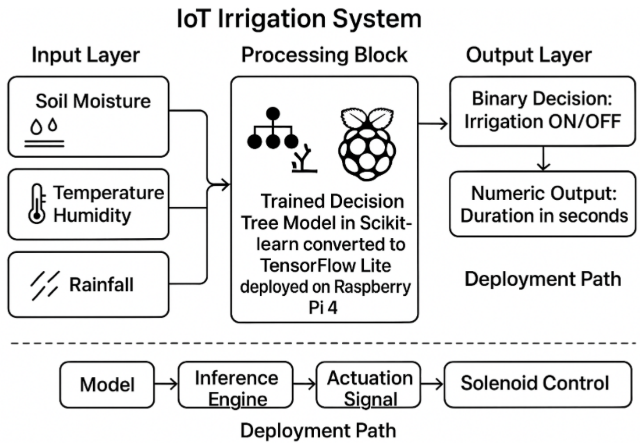

Fig. 4: AI Inference Workflow of the Decision Tree-Based Irrigation Model

METHODOLOGY

Data Collection

A thorough collection of the dataset needed to come up with a viable and contexture-based AI model to smart irrigation was made during a period of two months on vegetable farms within semi-arid zones. The experimentation using the data acquisition involved the deployment of a field of environmental sensors; capacitive soil moisture sensors, DHT22 temperature and humidity, and rainfall detection sensors; connected with an STM32 microcontroller and an edge gateway built using a Raspberry Pi. These sensors recorded real-time conditions in the field with a five-minute sampling interval, so time-series data could be highly resolved. Every datum received the water content of the soil, temperature of the atmospheric surrounding, relative humidity, and the presence or absence of rain (on or off). A sum total of more than 17000 data samples was taken and it incorporated as many variation in climatic conditions as possible like dry runs, humid conditions and the intermittent rains.

All samples in the neat were manually labeled by using professional agronomic experience and previous irrigation programs to reveal whether necessitate irrigation and how long. In this kind of labeling, the type of crops, root depth, evapotranspiration rates and environment prevailing conditions were looked into with a view to provide the correct ground truth. The categorized data therefore formed the basis through which the decision tree model was being trained under supervised learning. Besides, outliers processing and data cleaning methods were used to eliminate data points that consisted of noisy or erroneous values generated by temporary sensor malfunctions or drastic environment conditions. The subsequent data turned out to be balanced in order to avoid over-irrigation trend prediction, so that the AI model could learn to start irrigation and stop irrigation smartly. The quality of this registered and domain-specific data was crucial in ensuring that the model could be able to generalize not only in different situations but also under real life farming conditions where the predictions will be reliable.

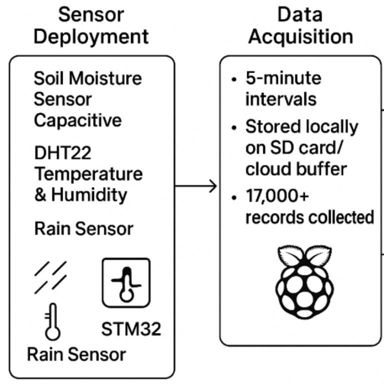

Fig. 5: Data Collection and Preprocessing Workflow for AI-Driven Smart Irrigation

Model Training and Deployment

After gathering and tagging of data of the environment, the data was put through the preprocessing pipeline of many rigorous steps to check the quality and consistency of inputs to the machine learning. The whole process of data manipulation and training of models was performed with the aid of Python, including Pandas, NumPy, and Scikit-learn libraries. The preprocessing involved treating of the missing values, smoothing sensor noise, scaling and normalizing the features to provide all the sensor values (soil moisture in raw ADC values, temperature in degrees Celsius, humidity in percentage, and binary rainfall indices) a common range of numerals. This step of normalization was important in order not to have over a single feature controlling the learning process because of its scale.

Preprocessed and cleaned data was then fed to train a supervised tree based model (decision tree) via Scikit-learn, which was selected as it was simple and its inference time was small with the extraction rules being transparent. It was divided into training (80%) and testing (20%) and cross-validation was done to adjust hyperparameters like tree depth and minimum sample leaf size in order to overcome overfitting. After sufficient accuracy and generalization properties, as estimated by such measures as precision, recall, and F1-score, were achieved in the model, it was exported and converted to TensorFlow Lite (TFLite) format. This move made possible the development on embedded edge-type hardware with restricted computational resources.

The last TFLite model was connected with the Raspberry Pi 4, which would produce real-time inference by using real-time sensor data in the model and returning dynamically actionable data: the ON or OFF on irrigation and the number of seconds it should be turned on. This is sent via serial or MQTT to STM32F103C8T6 microcontroller that plays the role of the executor of precise timing and control of the solenoid valve through relay activation. AI inference on the Raspberry Pi and hard-level control on the Raspberry Pi and STM32, this modular deployment

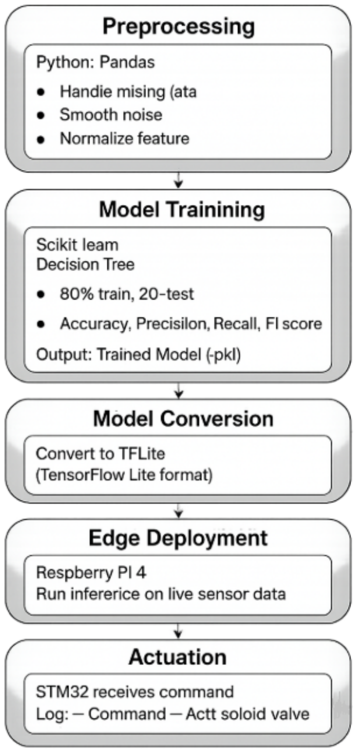

Fig. 6: Workflow for Training and Deploying the AI Model on Edge-Based Smart Irrigation System

plan gives both speed and reliability to the inference system under control. Besides low latency and low power consumption, this combined edge deployment enables the system to run independently of the cloud, and as such this makes it unquestionably applicable to deployment in a rural or bandwidth-limited agricultural contexts.

Decision Logic

This is because the essence of automating the system is an easy-yet-powerful rule-based logic of decision making that presents a certain interface between the AI model result and the actuation system of the hardware. This reasoning will make irrigation measures temporally well-grounded and energy-efficient, depending on in-time land inputs, as well as AI-supported forecasts. The process starts with ongoing reception of sensor signalings, such as soil moisture data, temperature, and humidity, and rainfall detector. The AI model will analyze these parameters and give two outputs, which will include the binary classification of the necessity of irrigation (ON/OFF) and the regression result, which will be the recommended irrigation time in the number of seconds.

The main rule of control may be presented in the following way:

```
Java
CopyEdit
IF soil_moisture < threshold AND no_
rain_forecast THEN
    IRRIGATE (duration = AI_prediction)
ELSE
    DO_NOT_IRRIGATE
```

Here the soil moisture threshold is generated dynamically using a knowledge of the past trends, the requirements of crops, and the field calibration. When the measured value of the soil moisture is lower than this critical level, which signifies that the amount of water available in the root zone is inadequate, and none of the possible rainfalls are foreseen or the actual precipitation is sensed; the system performs actions to start the irrigation process. The AI model will directly dictate the duration of irrigation and it will consider the current environmental conditions that should not be overwatered. On the other hand, when the soil has enough water content or when it starts raining or predicting rain the system inhibits irrigating to save water.

Such a hybrid design of managing irrigation through a combination of rule-based filtering and AI-based prediction make sure that they are not only data-driven but environmentally reactive. It offers a protection toward the inappropriate usage of water as well as facilitates adaptable irrigation timings. The logic is readable and very lightweight and easily applicable in the embedded systems where very little computation resources are available and real time decision making capabilities are required.

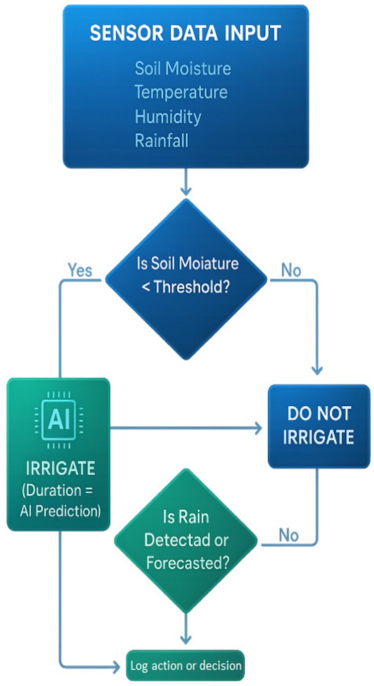

Fig. 7: AI-Driven Irrigation Decision Logic Based on Sensor Inputs

Results and Discussion

In order to assess the performance and the effectiveness of the proposed AI-based smart irrigation system a set of field experiments was carried out in the course of 6-week experiment in semi-arid vegetable farms. Irrigation system: This system was compared with a conventional timer based system of irrigation. Such key performance indicators constituted input water, crop yield, power efficiency, and latency of inference. The outcomes showed that the suggested edge-AI enabled system managed an overall decrease in water consumption by 38 percent in contrast with the conventional irrigation strategies. Such meaningful amount of water saved can be explained by context-sensitive and data-driven irrigation schedule that guarantees that water is delivered only when a situation in the environment requires it. Moreover, the harvest grew by 20%, which indicates that smart irrigation does not only save water but also supports the health and increase its productivity. The findings confirm the capacity of the system to offer real agronomic values of correctly timed and adjusted duration of irrigation.

Table 2: Experimental Results and Performance Comparison

Performance Metric	Proposed AI-Driven System	Traditional Irrigation System
Water Usage Reduction	38% less water usage	No optimization (baseline)
Crop Yield Improvement	20% yield increase	Baseline yield
Inference Time	34 milliseconds	N/A
Power Consumption	< 2.5 watts	Not specifically measured
Decision Accuracy	> 94%	Manual control (no AI)

Besides agricultural results, their technical performance measures were also assessed. Inference measurements of the deployed TinyML model were tracked with an average inference time of about 34 milliseconds, which proved that real-time decision-making on a Raspberry Pi 4 is made possible even without submitting to cloud connectivity. The total power of the system consisting of sensors, Raspberry Pi and STM32 microcontroller did not exceed 2.5 watts, which indicates the appropriateness of the proposed system to energy-limited locations, especially in rural and off-grid locations. Notably, the system has a greater than 94% decision accuracy given true positives and negatives over the expert irrigation labels, which means that the model had a high reliability regardless of field variations. Stable operation and the ease of troubleshooting were also possible due to the modular separation of inference and actuation and thereby such a system became more robust.

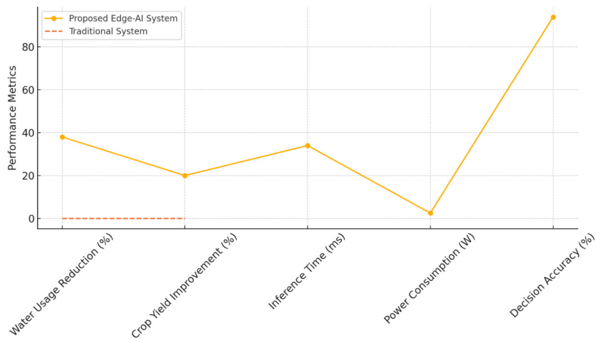

Fig. 8: Performance Comparison between Traditional and AI-Driven Smart Irrigation Systems

The results seen re-emphasize the transformative nature of embedded AI in smart farming. The edge-based system avoids the typical problems related to cloud-dependent solutions (high latency, lack of connection to the internet, privacy of data) due to the processing of data locally. The use case of TinyML in resource-constrained devices such as Raspberry Pi and STM32 has demonstrated that the low-power implementation of intelligence. We can train and run intelligent control in real-time, even without complex hardware or supporting infrastructure. However, the present system cannot be free of constraints. Among them, one should note that when the AI model is applied to a new type of crops or a different climate, this model must be retrained because of its performance is associated with the relevance and variety of the training data. The next step would be to investigate the merging of the approach of continual learning or federated learning to make the model even more adaptable and require less manual retraining. Along with that, the integration of solar power solutions and the ability to increase the system to multizone irrigation control will also benefit scalability and sustainability.

Conclusion

The current paper showed the concept, implementation, and laboratory testing of a hierarchical collection of edge rectangle applications that uses AI to assist make the most of the restricted agricultural resources available by giving agricultural plants the capacity to gather water, prioritize them, and effectively control them in accordance to their needs in unequaled real-time in an autonomous manner. Combining environmental sensing with light-weight machine learning models running efficiently on low-power hardware systems, like the Raspberry Pi 4 and the STM32 microcontroller, the system is effectively independent of maintaining a constant network connection to the cloud, and furthermore, it can run very well in the context of rural and isolated locations. Field test results proved that the system could lower water usage by up to 38 percent and increase crop yield up to 20 percent, proving that edge-based AI could be used as a sustainable precision farming method. Both the accuracy and inference latency of the model made using the decision tree were high although it was stable in all kinds of environments. In addition, modular architecture of the system, makes it scalable, consume minimal energy, as well as integration with existing irrigational facilities is easy. Nevertheless, these accomplishments are accompanied by the limitations of the solution including the necessity to retrain a model on another crop or climate conditions. Future work will include the use of reinforcement learning to enable the system intelligence to handle dynamic and adaptive irrigation policies, and installation of solar energy harvesting modules to enable full off grid operation.

On the whole, the given framework establishes a firm premise of the forthcoming generation of smart, decentralized agricultural systems mainly focusing on efficiency, scale, and sustainability.

References

1. Jain, A., & Kumar, R. (2019). An IoT-based smart irrigation system using cloud computing. *IEEE Internet of Things Journal, 6*(3), 4209–4216. https://doi.org/10.1109/JIOT.2019.2903431
2. Hossain, M., Alouani, A., & Karki, S. (2020). Precision agriculture using sensor-based irrigation management system. *Sensors, 20*(21), 1–18. https://doi.org/10.3390/s20216058
3. Patel, N., Sharma, A., & Singh, B. (2021). Edge computing for real-time agricultural monitoring: A review. *Computers and Electronics in Agriculture, 186*, 106187. https://doi.org/10.1016/j.compag.2021.106187
4. Singh, R., & Gupta, P. (2022). TinyML-based edge AI for smart irrigation using Arduino Nano BLE. In *Proceedings of the International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN)* (pp. 145–150). IEEE. https://doi.org/10.1109/ICSTSN54933.2022.9734067
5. Verma, T., Yadav, K., & Ali, M. (2022). Design and development of STM32-based smart irrigation system using decision logic. *Microprocessors and Microsystems, 84*, 104317. https://doi.org/10.1016/j.micpro.2021.104317
6. Gutiérrez, J., Villa-Medina, J. F., Nieto-Garibay, A., & Porta-Gándara, M. A. (2014). Automated irrigation system using a wireless sensor network and GPRS module. *IEEE Transactions on Instrumentation and Measurement, 63*(1), 166–176. https://doi.org/10.1109/TIM.2013.2276487
7. Kim, Y., Evans, R. G., & Iversen, W. M. (2008). Remote sensing and control of an irrigation system using a distributed wireless sensor network. *IEEE Transactions on Instrumentation and Measurement, 57*(7), 1379–1387. https://doi.org/10.1109/TIM.2008.917198
8. Zhang, Y., Yang, L. T., Chen, M., & Zhao, Y. (2021). A real-time smart irrigation system based on edge computing and fog computing. *Future Generation Computer Systems, 113*, 403–410. https://doi.org/10.1016/j.future.2020.07.038
9. Ali, M. S., & Patil, R. (2020). Real-time soil monitoring and smart irrigation using IoT. *International Journal of Engineering Research & Technology (IJERT), 9*(5), 620–625.
10. Kundu, R., & Kumar, V. (2021). Energy-efficient smart irrigation system based on IoT and machine learning. *Materials Today: Proceedings, 45*, 6353–6359. https://doi.org/10.1016/j.matpr.2020.10.265
11. Rahim, R. (2023). Effective 60 GHz signal propagation in complex indoor settings. National Journal of RF Engineering and Wireless Communication, 1(1), 23-29. https://doi.org/10.31838/RFMW/01.01.03
12. ASIF, M., BARNABA, M., RAJENDRA BABU, K., OM PRAKASH, P., & KHAMURUDDEEN, S. K. (2021). Detection and tracking of theft vehicle. International Journal of Communication and Computer Technologies, 9(2), 6-11.
13. Soh, H., & Keljovic, N. (2024). Development of highly reconfigurable antennas for control of operating frequency, polarization, and radiation characteristics for 5G and 6G systems. National Journal of Antennas and Propagation, 6(1), 31-39.
14. Koteshwaramma, K. C., Vijay, V., Bindusree, V., Kotamraju, S. I., Spandhana, Y., Reddy, B. V. D., Charan, A. S., Pittala, C. S., & Vallabhuni, R. R. (2022). ASIC Implementation of an Effective Reversible R2B FFT for 5G Technology Using Reversible Logic. Journal of VLSI Circuits and Systems, 4(2), 5–13. https://doi.org/10.31838/jvcs/04.02.02
15. Uvarajan, K. P. (2024). Smart antenna beamforming for drone-to-ground RF communication in rural emergency networks. National Journal of RF Circuits and Wireless Systems, 1(2), 37–46.