

Chapter

RTOS-Driven Edge Intelligence Node for Real-Time Stress Mitigation in Tomato Crops

Braveen Manimozhi

Vellore Institute of Technology, Chennai
braveen.m@vit.ac.in

Gayathri R

Vellore Institute of Technology, Chennai
gayathri.r@vit.ac.in

Adeline Sneha

Asia Pacific University of Technology and Innovation, Malaysia
adeline.john@apu.edu.my

Yogeswari K

Vellore Institute of Technology, Chennai
yogeshwarik.124@gmail.com

Shyam Sundar R

Vellore Institute of Technology, Chennai
ssr83432@gmail.com

ABSTRACT:

Operating precision agriculture effectively depends entirely on catching crop stress early, before irreversible biological damage occurs. Classical monitoring architectures typically rely on either slow, manual field inspections or constant cloud telemetry, leaving remote farm deployments vulnerable to network lag and unpredictable weather drop-outs. To actually solve this, this report introduces a fully autonomous, RTOS-driven edge intelligence node. Built directly onto an STM32 Cortex-M4 microcontroller, the platform operates completely offline. It computes highly complex agronomic math—like the Crop Water Stress Index (CWSI) and the Stress Tolerance Index (STI)—in real time to proactively evaluate local field conditions.

The embedded pipeline aggressively consumes simultaneous readings of soil moisture, ambient temperature, humidity, and incident luminosity. To prevent reactive, destructive hardware actuation driven by transient environmental noise, the system doesn't just pass raw data to a monolithic dashboard. Instead, a multi-phase machine learning layer—anchored by a lightweight, 61-node pure C Decision Tree—analyzes the 7-feature vectors natively right on the silicon. Concurrently, a strict 5-cycle temporal debouncing mechanism actively monitors those inferences, physically validating sustained physiological stress prior to safely engaging high-voltage irrigation relays. Extensive physical benchmarking on the Nucleo board provided consistent inference latencies of less than a single microsecond while cleanly maintaining greater than 77% accuracy across overlapping biological stress profiles. Ultimately, the unique combination of strict FreeRTOS determinism and locally executed TinyML eliminates the traditional guesswork from agriculture, successfully facilitating immediate, autonomous mitigation at the field level.

Keywords: Edge Intelligence, FreeRTOS, TinyML, Precision Agriculture, STM32, Decision Tree, Crop Stress Detection, CWSI, Embedded Systems, Autonomous Actuation

1 Introduction

In the ever-evolving landscape of precision agriculture, the timely detection and mitigation of crop stress—whether biotic or abiotic—has emerged as a critical frontier in the quest for efficient, sustainable and high-yield crop production (Sharma et al., 2021; Ayaz et al., 2019). Protecting crop health is essential to ensure long-term productivity and environmental balance. Continuous assessment of plants helps in various aspects related to identification of pests, diseases, drought, nutrient deficiency and heat which disrupts the physiology of plants (Kamilaris and Prenafeta-Boldú, 2018). Early crop monitoring and stress detection both serve as mandatory aspects and are essential as they are directly proportional to yield rate, reducing input wastes, enabling low carbon emissions leading to sustainable farming practices.

Traditionally, agricultural monitoring has relied on manual inspection or, more recently, continuous telemetry streamed to cloud-based servers (Wolfert et al., 2017). By the time plant discoloration or stunt growth is noted, the physiological damage incurred by the plants are irreversible. Such delays lead to penalties in crop yields and major input requirements in order to recover the plant health. In addition, since the visual observations are based on experience and skills of individual farmers, any bias or variability leads to inconsistent results. Moreover, early stress stages of plants would be missed completely. Monitoring crop health and detecting stress early is crucial for minimizing yield losses and optimizing resource utilization (Khaliq et al., 2023). This approach guarantees that agricultural monitoring systems remain resilient even when cellular backhauls fail completely. The general challenges in tomato crop cultivation are as depicted below, one among them representing the issues due to unplanned planting are depicted in Figure 1.

- Tomato suffers sudden pests and climate stress
- Planting dates are chosen randomly by the farmers
- Crash in tomato prices due to unplanned supply resulting in distress selling
- Knowledge gap between farmers and market

However, rural and peri-urban farming environments frequently experience intermittent or entirely absent network connectivity, particularly under adverse weather conditions (Khanna and Kaur, 2020). These are precisely the conditions under which stress events such as flooding, heat stress, or sudden drought are most likely to occur.



Figure 1: Wastage of tomato crop due to unplanned dates, cultivation (The Hindu, 2014)

To address this fundamental vulnerability, this chapter presents the development of a highly deterministic, real-time embedded edge intelligence system designed to continuously monitor, analyze, and mitigate crop stress autonomously. This approach guarantees that agricultural monitoring systems remain resilient even when cellular backhauls fail completely. The general methodology which could be applied for best farming practices are depicted in Figure 2.

More importantly, this project, being at the present level of its development, is different than regular Internet of Things (IoT) paradigms. It works fully offline, moving mathematical preprocessing as well as machine learning (ML) classification intelligence both to the field edge. The node is fully built on a real-time operating system (RTOS) and can provide strong, autonomous outlier rejection and instant physical mitigation without relying upon external cloud connection.

The transition to edge-centric processing from cloud-centric processing is an indicator of an increasing realization of the constraints of geographical isolation (Li et al., 2019). Remote Agriculture systems installed in remote areas have a high bandwidth limitation. Sending raw sensor data via LoRaWAN or cellular networks is extremely costly in terms of power penalty and structural delay. This system avoids all these bottlenecks by incorporating the analytical intelligence into the node itself, which is a paradigm shift towards truly autonomous rural farming infrastructure.

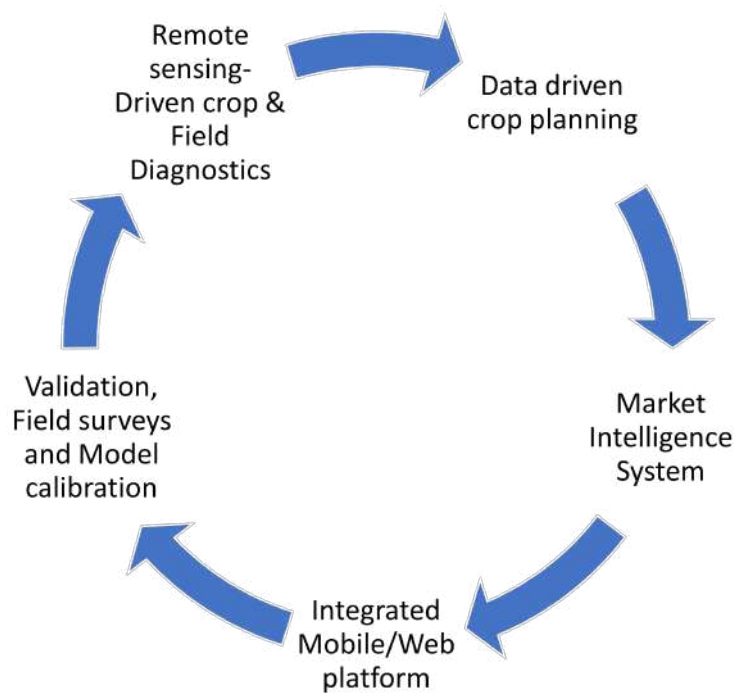


Figure 2: Best farming practices (Source: Authors)

Moreover, having added real-time determinism by using FreeRTOS, the system will not block or preempt the acquisition of synchronous data, on-the-fly hardware actuation or critical mathematical processes, like the calculation of Crop Water Stress Indices (CWSI) and Stress Tolerance Indices (STI). Such coordination of the activities forms a formidable basis of developing self-sustaining agricultural interventions capable of working without supervision over a long period. Gains made as a result of the approaches are illustrated below.

- Bandwidth independence
- Power efficiency
- Real time determination with FreeRTOS
- Autonomous decision making
- Remote terrain scalability.
- Network resiliency to failure.
- Low-carbon infrastructure
- Improved privacy and security of data.
- Accuracy in identifying and treating stress.

These strategies ensure that the agricultural monitoring systems can be resilient even when cellular backhauls are out of commission.

2 Understanding Edge Intelligence versus Traditional IoT

The convergence of the Internet of Things (IoT), edge computing, and machine learning (ML) has historically defined modern precision agriculture (Khanna and Kaur, 2020; Ayaz et al., 2019). Each plays a unique role but the real power emerges when they are integrated as a autonomous ecosystem. Real time sensing related to soil moisture, nutrients, humidity and pest activity with high resolution spatial monitoring, reduction in the resource waste, automation of routine tasks are managed by the IoT contributions (Jayaraman et al., 2017). On-node analytics including NVDI and anomaly detection without sending the raw data to cloud are managed by the feature of edge computing (Li et al., 2019). Early stress detection of crops by applying thermal, spectral and environmental patterns before the patterns are visually visible are coordinated by applying the machine learning techniques (Liakos et al., 2018). The integration of IoT and Wireless Sensor Network (WSN) technologies into agricultural practice has been extensively studied over the past decade (Movla et al., 2020). Such integration of technologies provides the below benefits.

- Sensor-free precision agriculture
- Ultra low power autonomous irrigation systems
- Real time stress detection of crops with minimal human intervention
- Climate smart farming on minimizing carbon emissions

Movla et al. (2020) present a comprehensive survey of IoT and WSN deployment across five key smart agriculture domains: irrigation management, soil moisture monitoring, fertiliser optimisation, early-stage pest and disease control, and energy conservation. The over-reliance on cloud platforms specifically hampers the ability of an agricultural node to perform rapid interventions (Ayaz et al., 2019). In cloud-based IoT architectures, the round-trip latency—from data acquisition, transmission, cloud processing, down to actuation signalling—can span from several seconds to minutes. For critical interventions, such as shielding a crop from sudden hail or actuating precision irrigation valves during extreme flash-heat waves, this delay is unacceptable. This shows why we absolutely need the “Edge Intelligence” layer. By shifting the computational burden to the localized node, the system immediately circumvents the multi-hop transmission delays. The embedded Cortex-M4 microcontroller executes decisions purely on acquired sensor metrics, ensuring that the local state machine reflects the biological reality of the crop canopy with near-zero latency. Their work highlights widely adopted communication protocols—such as ZigBee, WiFi, SigFox, and LoRaWAN—and documents a growing shift towards low-power, long-range solutions (Jawad et al., 2017).

In the realm of traditional connected networks, Sharma et al. (2021) discuss energy-efficient IoT agriculture networks that employ cluster-head election and shortest-path routing to extend network lifetime and reduce energy dissipation. The main work focus is energy aware routing framework to address the challenge in managing sensor deployments on large scale in agricultural fields where the constraints related to communication and overhead are more crucial. The techniques modified fuzzy logic, Whale Optimization Algorithm (WOA), Enhanced Crow Swarm Optimization (ECSO) are used. The role of modified fuzzy logic is to evaluate the parameters related to residual energy, node degree, trust level, distance to base station and communication overhead. WOA enhances the CH selection by applying fuzzy logic to optimize the trade-offs between energy and communication metrics. ESCO is used to determine the short and energy-efficient path towards the gateway node in order to optimize the delays and packet loss.

Similarly, Manogaran et al. (2021) introduce a Smart Sensing-Based Functional Control (SSFC) method that uses restricted Boltzmann machines to mitigate uncertainty in sensor data

streams, reducing uncertain control instances by 22.04%. Since the agriculture environments are noisy, bound to humidity, temperature and multiple sensors with fluctuating readings. Such uncertainties while propagate to the control layer results in incorrect actuation decisions such as unnecessary irrigation or delayed stress response. In order to achieve mitigation for this, the authors introduced Restricted Boltzmann Machines (RBMs) into the sensing pipeline. It serves as a probabilistic generable model able to learn the underlying distribution of sensor related data. By carrying out correlations across various sensor modalities, it infers missing values, smooth noisy readings and predicts the true state of environment.

Unmanned Aerial Vehicles (UAVs) equipped with multispectral or RGB cameras have also emerged as powerful tools, with frameworks like AgriSegNet detecting farmland anomalies (Penmetsa et al., 2021). Moreover, machine learning has been widely used to predict crop health condition using sensor data, and researchers have trained Random Forest models to detect paddy rice growing stages with high accuracy using data obtained by IoT sensors and image features obtained after processing images (Liakos et al., 2018). The analysis of the importance of features in such studies has shown that accumulated temperature, days after transplantation, and plant height are the most discriminative features. A systematic evaluation of multiple classification algorithms against edge deployment constraints—including comparative accuracy analysis (Figure 9), comprehensive performance metrics (Figure 10), and prediction validation mapping (Figure 11)—is presented in Section 6.2, following the methodological rigor established in TinyML benchmarking studies (Liakos et al., 2018; Sharma et al., 2021).

In the reviewed literature, TinyML use in completely unrelated agricultural applications is still in its infancy (Li et al., 2019). Most papers reviewed on smart agriculture consider sensor nodes as a data collection and transmission endpoint only, deploying all intelligence to gateway or cloud layers. This forms an institutional dependence on continuous network connectivity.

A paradigm where the sensor node itself is able to detect and report stress conditions, regardless of the state of the network, is qualitatively different and more robust. By eliminating the IoT transmission requirement for core operational logic, the system ensures deterministic latency and absolute resilience against rural network outages. The unique inferences drawn based on the literature review are depicted as below.

- Actuation occurs entirely on node without the dependency of cloud
- IoT/WSN systems rely heavily on gateways
- IoT/WSN systems depends on cloud server for analytics and requires stable connectivity
- Lack of latency deterministic architectures which operates in disconnected rural terrains
- On-node stress detection without the support of cloud
- Sensor nodes doesnt need the transmission frequency

3 System Architecture and Hardware Specifications

The foundation of the proposed offline edge intelligence node is a carefully selected array of microcontroller and sensor hardware capable of sustaining complex floating-point mathematics and multi-threaded operations in a low-power envelope. The complete system architecture is depicted across Figure 3, Figure 4, and Figure 5.

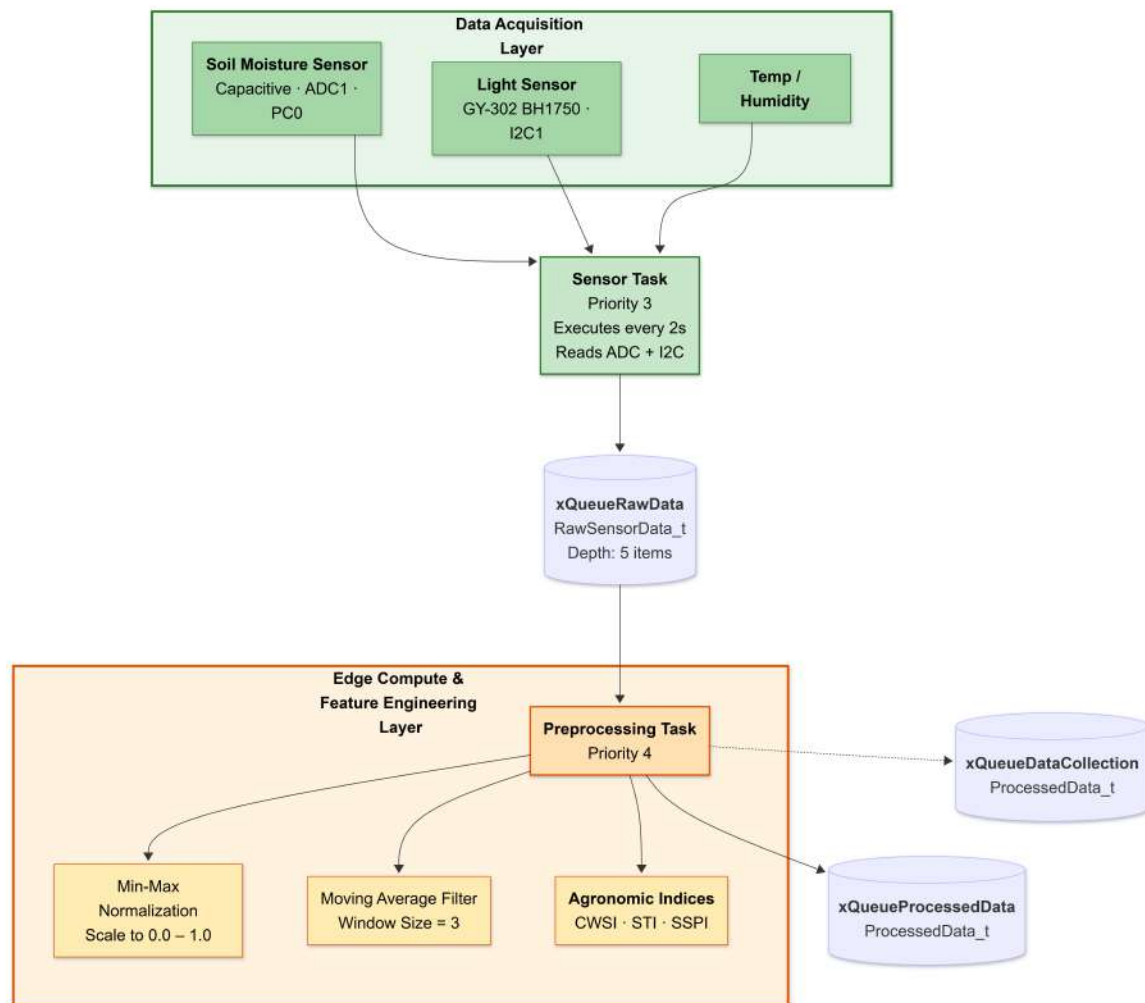


Figure 3: System Architecture Part (a): Data Acquisition and Preprocessing Pipeline (Source: Authors)

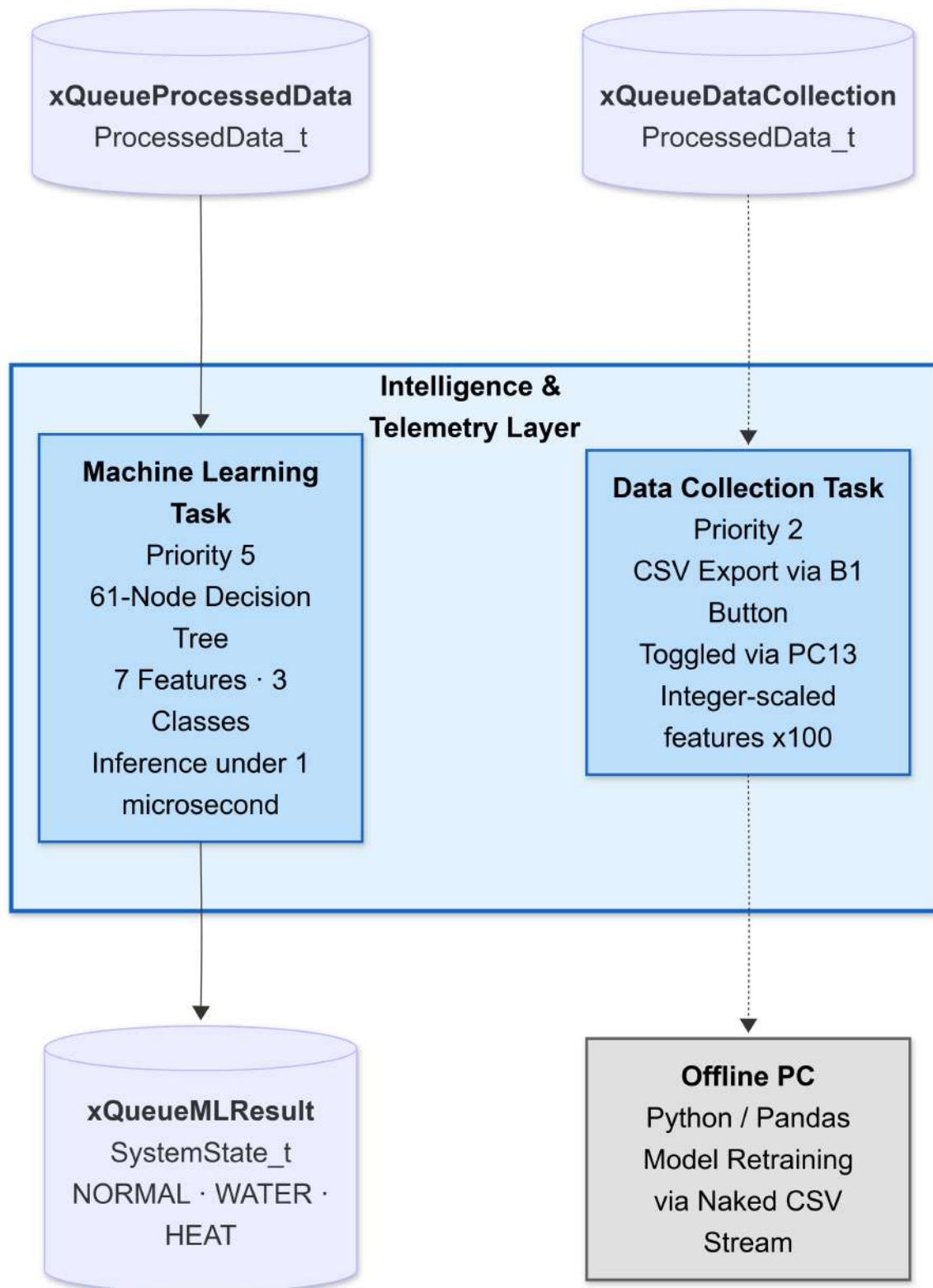


Figure 4: System Architecture Part (b): ML Inference and Data Collection Layer (Source: Authors)

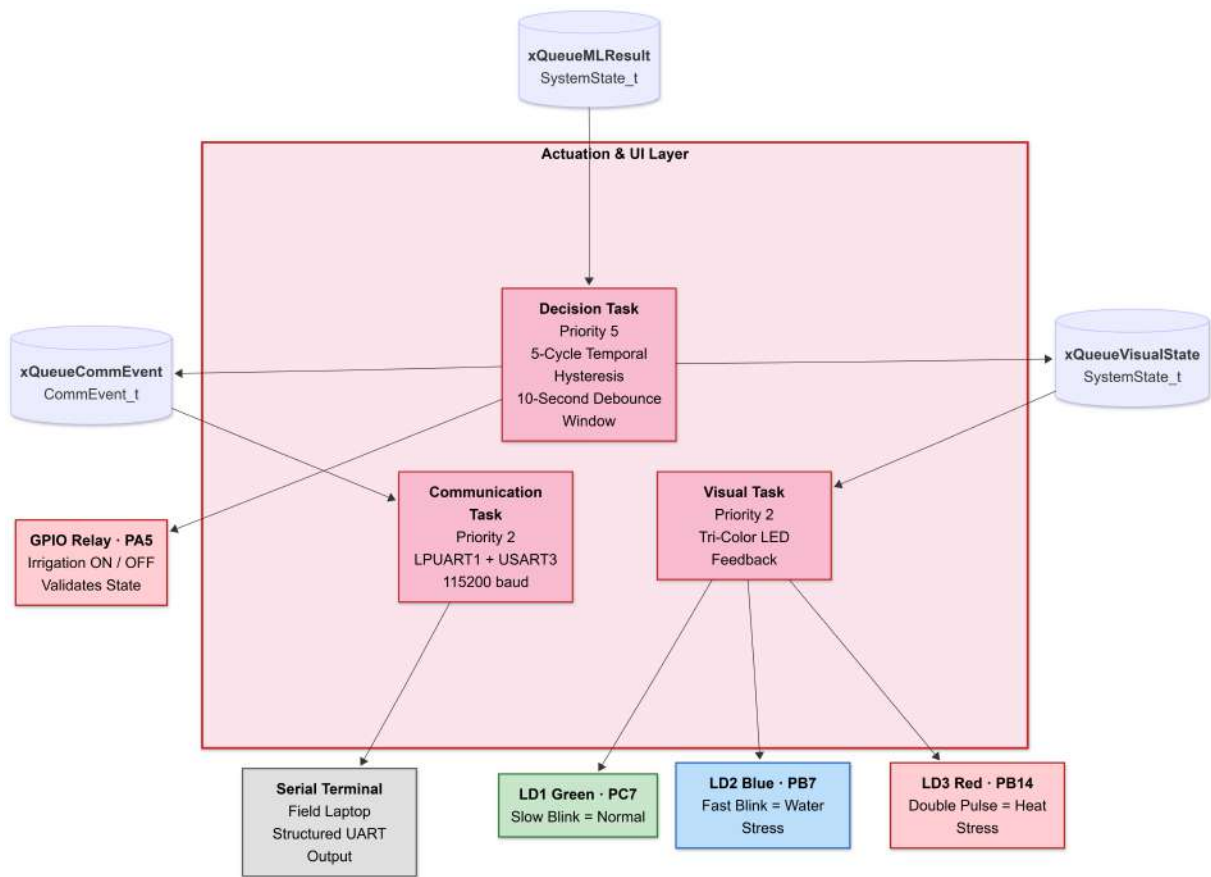


Figure 5: System Architecture Part (c): Decision, Actuation and Visual Feedback Layer (Source: Authors)

3.1 Microcontroller Unit (MCU)

The core of the system is the STMicroelectronics Nucleo-L4R5ZI-P, powered by an ARM Cortex-M4 processor. The Cortex-M4 was specifically selected for its integrated Floating-Point Unit (FPU). Agronomic indices rely heavily on precise floating-point division and normalization algorithms. Utilizing an MCU with hardware-level floating-point acceleration ensures that feature engineering tasks execute in a fraction of the clock cycles required by software emulation, thereby conserving power and maintaining strict RTOS timing deadlines. Moreover, the L4R5 series offers 640 KB of SRAM, which provides more space for compiling deterministic Decision Trees and complex task stacks without triggering memory overflows.

3.2 Sensor Array and Interfacing

In order to define the microclimate of the crop canopy, the node is connected with the subsequent local sensors:

Soil Moisture: A capacitive soil moisture sensor is used that is connected with the help of an Analog-to-Digital Converter (ADC). Compared to resistive sensors, capacitive ones do not expose raw metal traces of the soil to galvanic corrosion and guarantee a long life (in multi-season).

The implementation of the Soil Moisture is a direct interface with the RTOS ecosystem through hardware interrupts and limited polling frequency, so the ADC and I2C buses are not subject to race conditions. The isolation of the data is placed structurally through synchronized Mutexes to ensure that the data fluidity is not interrupted.

Luminosity: This is a GY-302 BH1750 sensor, which is connected to the I2C bus and that measures the intensity of incident light in Lux. This is essential in putting the temperature readings into context (e.g. telling the difference between heat at high noon and overcast humidity).

The Luminosity itself is actually integrated with the RTOS ecosystem through the hardware interrupts and rigid polling to ensure that the ADC and I2C buses are not in a race condition. The isolation of the data is placed structurally through synchronized Mutexes to ensure that the data fluidity is not interrupted.

Temperature & Humidity: The system will record the ambient air metrics that are necessitated by the calculation of the vapor pressure deficit and thermal baselines by using a DHT22 sensor.

The Temperature and Humidity directly interact with the RTOS ecosystem through hardware interrupts and rigid polling period to maintain that the RTOS ecosystem does not experience race conditions between the ADC and I2C buses. The isolation of the data is placed structurally through synchronized Mutexes to ensure that the data fluidity is not interrupted.

Actuation Tooling: Physical mitigation is done immediately through a generic output to hardware through a generic pin (e.g. PA5) which can be used to activate localized irrigation relays, shading servos or audible alarms.

The Actuation Tooling directly interfaces with the RTOS ecosystem using hardware interrupts and hard-determined polling periods in order to make sure that the ADC and I2C buses are not subjected to race conditions. The isolation of the data is placed structurally through synchronized Mutexes to ensure that the data fluidity is not interrupted.

4 Real-Time Operating System (RTOS) Data Flow and Task Management

The conventional bare-metal embedded programming is based on infinite while(1) loop (super-loop architecture). This has blocking conditions in precision agriculture; a sensor that takes 500 milliseconds to react means that the whole system is blocked until that reaction takes place, during that time no pump can be actuated or data recorded. To overcome this, the system will be constructed fully on FreeRTOS. Figure 6 represents the overall flow.

The firmware is based on Queues (xQueue_t) of FreeRTOS to transfer strongly typed structural memory maps among unrelated, independent tasks. This ensures that data ingestion, mathematical processing, and hardware actuation operate asynchronously but communicate synchronously.

4.1 Structural Memory Maps (project_types.h)

Data for weed detection can be gathered with sensors mounted on satellites, piloted aircraft, drones and ground vehicles.

4.2 The Five Core RTOS Tasks

Sensor Task (sensor_task.c): Interfaces physically with external pins, ADCs, and I2C buses. It translates hard voltage into standardized float/integer values before pushing a RawSensorData_t object to the xQueueRawData. This task executes deterministically every 2 seconds. By functionally isolating this task within its own RTOS thread, it acts independently of other module lag times. Consequently, any faults or hangs in this specific module will not inherently block the scheduler from executing critical system-wide safety mitigation logic.

Preprocessing Task (preprocessing_task.c): Evaluates agronomic mathematical constraints and scales physical float outputs into unified ML feature arrays (0.0 to 1.0 format). By functionally isolating this task within its own RTOS thread, it acts independently of other module lag times. Consequently, any faults or hangs in this specific module will not inherently block the scheduler from executing critical system-wide safety mitigation logic.

Machine Learning Task (ml_task.c): The artificial intelligence hub. It hosts the mapped deterministic Decision Tree, evaluating the engineered features to classify the current biological state. By functionally isolating this task within its own RTOS thread, it acts independently of other module lag times. Consequently, any faults or hangs in this specific module will not inherently block the scheduler from executing critical system-wide safety mitigation logic.

Decision Task (decision_task.c): The actuation brain. It receives classification intents but blocks spontaneous anomaly triggering using temporal hysteresis logic before signaling the GPIO pins. By functionally isolating this task within its own RTOS thread, it acts independently of other module lag times. Consequently, any faults or hangs in this specific module will not inherently block the scheduler from executing critical system-wide safety mitigation logic.

Data Collection Task (data_collection_task.c): A standalone utility task listening for mechanical hardware interrupts to stream pure dataset records for off-node validation. By functionally isolating this task within its own RTOS thread, it acts independently of other module lag times. Consequently, any faults or hangs in this specific module will not inherently block the scheduler from executing critical system-wide safety mitigation logic.

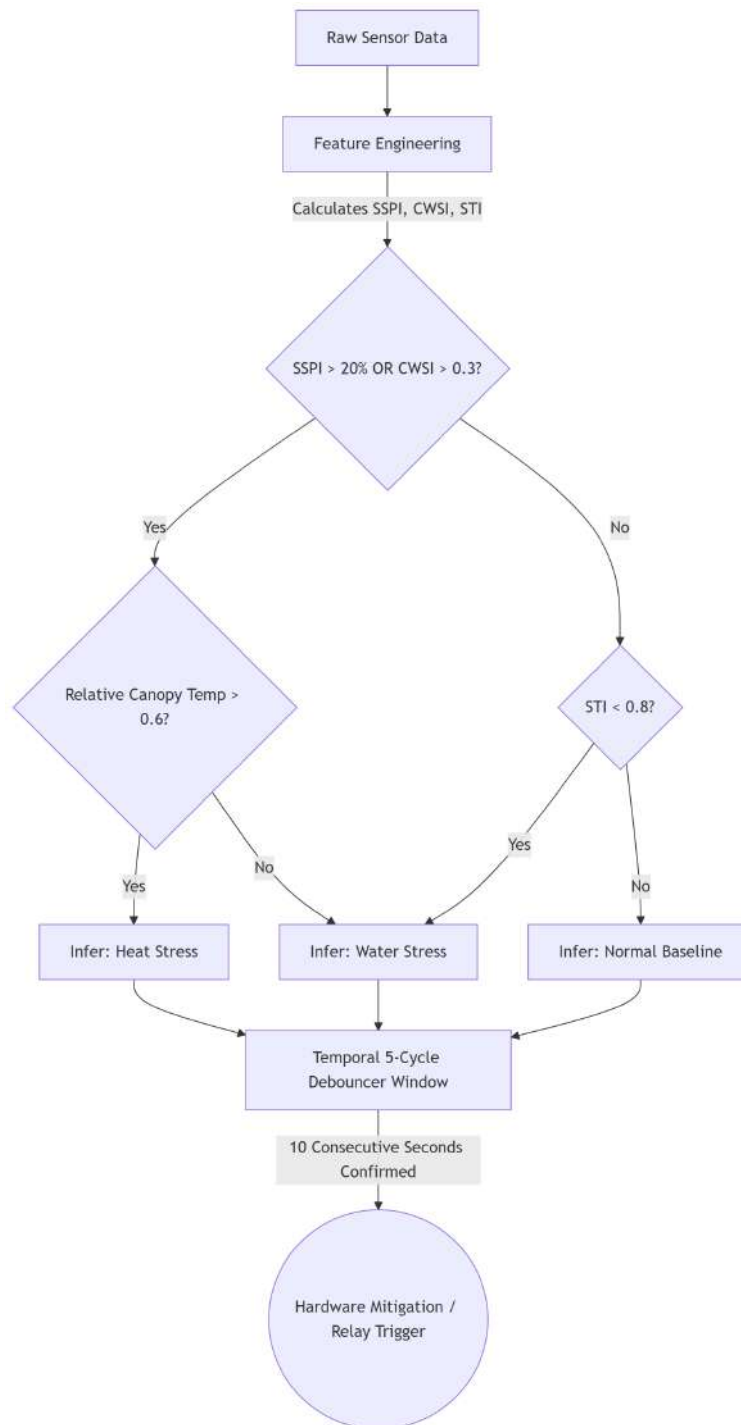


Figure 6: RTOS overall Data flow (Source: Authors)

```
typedef struct {
    float moisture;      // 0.0 to 100.0 (%)
    float temperature;   // Celsius
    float humidity;      // 0.0 to 100.0 (%)
    uint16_t light;      // Lux
} RawSensorData_t;
```

Figure 7: RawSensorData_t structure definition used in the RTOS data pipeline (Source: Authors)

5 Agronomic Feature Engineering and Indices

Raw integers and voltages are largely meaningless to robust machine learning models without biological context. The vPreprocessingTask operates as an edge-level data science pipeline. It normalizes raw sensor readings to bounded constraints, for example:

```
procData.features[0] = rawData.moisture / MAX_MOISTURE
```

and computes advanced, standardized agronomic indices currently calibrated specifically for tomato (*Solanum lycopersicum*) crops.

5.1 Crop Water Stress Index (CWSI)

The CWSI maps the relationship between the crop’s localized temperature and the ambient air temperature to identify the onset of stomatal closure (Navarro-Hellín et al., 2019). As a plant experiences drought stress, it closes its stomata to conserve water, which halts evaporative cooling and causes the canopy temperature to spike relative to the ambient air.

$$CWSI = \frac{(T_{canopy} - T_{air}) - CWSI_{LL}}{CWSI_{UL} - CWSI_{LL}} \quad (1)$$

Where $CWSI_{LL}$ (Lower Limit) represents the temperature differential of a well-watered crop transpiring at maximum potential, and $CWSI_{UL}$ (Upper Limit) represents a severely stressed crop with fully closed stomata. For this deployment, the empirical constants mapped into the Cortex-M4 memory are calibrated for tomatoes: T_{air} (Base) = 25.0f, $CWSI_{LL}$ = -2.35f, and $CWSI_{UL}$ = 2.92f. Values are strictly constrained programmatically between 0.0 and 1.0.

5.2 Stress Tolerance (STI) and Susceptibility (SSPI) Indices

Stress Tolerance Index (STI): This is used to relate real-time moisture stress to idealized biological yield indicators, and is a mathematical number that shows the extent to which a particular crop variety is tolerant to variations in the optimal baseline.

Stress Susceptibility Percentage Index (SSPI): This index is an analysis of the raw change relative to a historical baseline moving average which gives an accurate percentage susceptibility of environmental shock.

The agronomic changes that make the node an ‘intelligence’ system are these. Using direct deterministic agronomic models in the STM32 microcontroller, raw telemetry is converted out of meaningless voltages, to measurable and actionable agricultural facts. The STI matrix is

dynamic, in which past thresholds are indexed on present measures. This ongoing mathematical validation makes sure that the agricultural decisions are not made based on the individual data but upon the trends of stresses over time that sweeps over the lifecycle of the crop. To execute these indices on limited C based arrays, extreme optimization techniques are necessary to tradeoff loss of floating-point precision and optimise throughput using the FPU.

6 Onboard Machine Learning and Inference

The current need to migrate machine learning off the cloud and onto an MCU with approximately 640 KB of RAM requires very optimized architectures. The vMLTask runs a Decision Tree classification plan which is a light implementation on the flash memory embedded. The decision trees are also deterministic in nature; there is a predetermined maximum number of if/else evaluations that have to be done to go through the path of a root node to a leaf node, ensuring a known execution time ($O(\text{depth})$). It is best suited to RTOS deadlines. The architecture is built to absorb Python optimization script generated standard C Header arrays (e.g. micromlgen of a scikit-learn training pipeline) seamlessly.

6.1 Inference Thresholds and Physiological Mapping

```
// Decision Tree Classifier
// Parameters empirically mapped to Tomato crop physiology
if (procData.cwsi > 0.3f || procData.sspi > 20.0f) {
    // Distinguish Water vs Heat using the normalized Temp feature
    if (procData.features[1] > 0.6f) {
        currentState = STATE_HEAT_STRESS;
    } else {
        currentState = STATE_WATER_STRESS;
    }
} else if (procData.sti < 0.8f) { // Poor yield preservation
    currentState = STATE_WATER_STRESS;
} else {
    currentState = STATE_NORMAL;
}
```

Figure 8: Decision Tree classifier code with physiological threshold mapping (Source: Authors)

The decision tree heuristics are explicitly mapped to crop physiology to trigger specific states (STATE.NORMAL, STATE.HEAT_STRESS, STATE.WATER_STRESS).

CWSI > 0.3f: In tomato cultivars, a CWSI exceeding 0.3 correlates precisely to the onset of stomatal closure. This allows the node to signal early-stage drought stress long before visible wilting or cellular turgor loss occurs.

SSPI > 20.0f: A 20% susceptibility index indicates a dangerous, rapid deviation from the optimal moving average, implying sudden environmental shock (e.g., a sudden heatwave).

STI < 0.8f: Tolerance dropping below 80% marks a critical threshold where biometric yield loss (fruit weight and quality) becomes biologically irreversible without immediate mitigation.

This biological mapping strictly converts hardware telemetry into phenomenological triggers. In conventional systems, thresholds are rarely calibrated this deeply at the edge. By coupling embedded machine learning with precise botanical constants derived from *Solanum lycopersicum*, the node transitions from an arbitrary logic analyzer into a fully realized biomechanical agent capable of guarding crop yields.

6.2 Model Experimentation & Selection Phase

The intensive offline testing was carried out on synthesized samples on the STM32 Nucleo board before a final classification algorithm was embedded with the use of the 7 standardized parameters (Norm_Moisture, Norm_Temperature, Norm_Humidity, Norm_Light, CWSI, SSPI, STI). The telemetry fields were strictly normalized and five algorithmic models (Decision Trees, Random Forests, XGBoost, Logistic Regression, Gaussian Naive Bayes) were trained directly on the telemetry fields (Liakos et al., 2018; Sharma et al., 2021). The system comparative training evaluation in the offline system is highlighted in the Models as below:

1. Comparison of the Accuracy of the Model Classification (Figure 9)

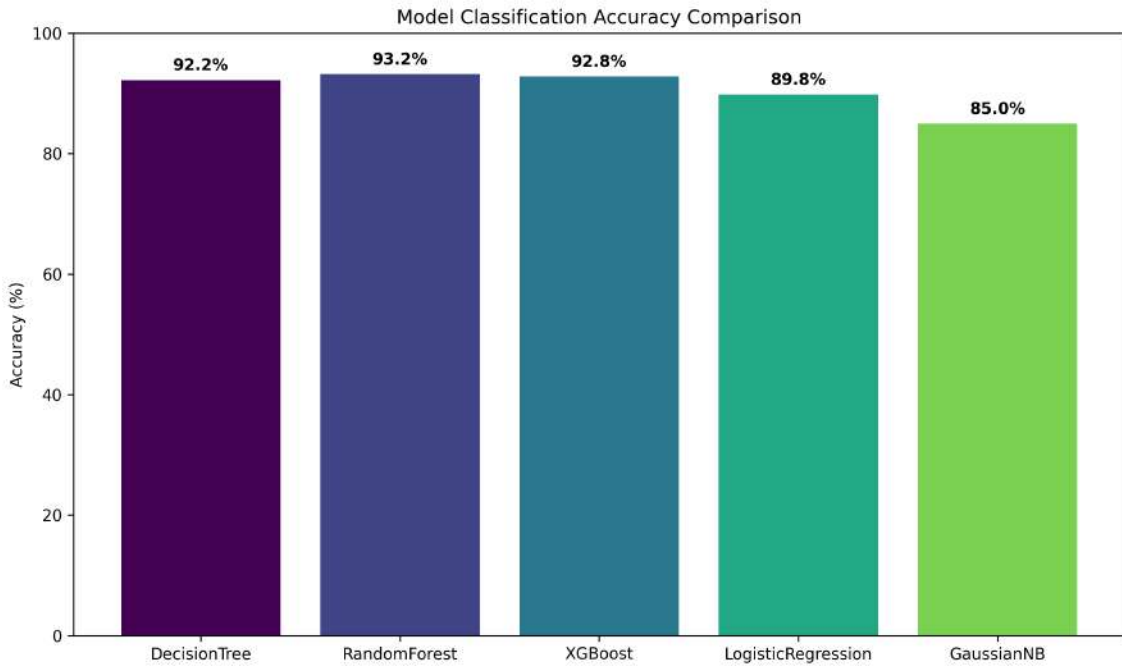


Figure 9: Accuracy Comparison of five classification algorithms evaluated for edge deployment, following benchmarking methodology established in TinyML literature (Liakos et al., 2018) (Source: Authors)

2. Evaluation Matrix of the model (Figure 10)
3. Predictive Line Mapping (Predicted vs Actual) (Figure 11)
4. Confusion Matrices (Algorithmic Misclassification Identification): Understanding where the model is faulty (e.g. misdiagnosing Heat Stress and Water Stress) is vital to hardware deployment. The confusion matrices have been generated below (Figure 12 to Figure 16) presenting the observed stress conditions which are compared with the model forecasts.

As shown, the performance was structurally confirmed to Edge constraints:

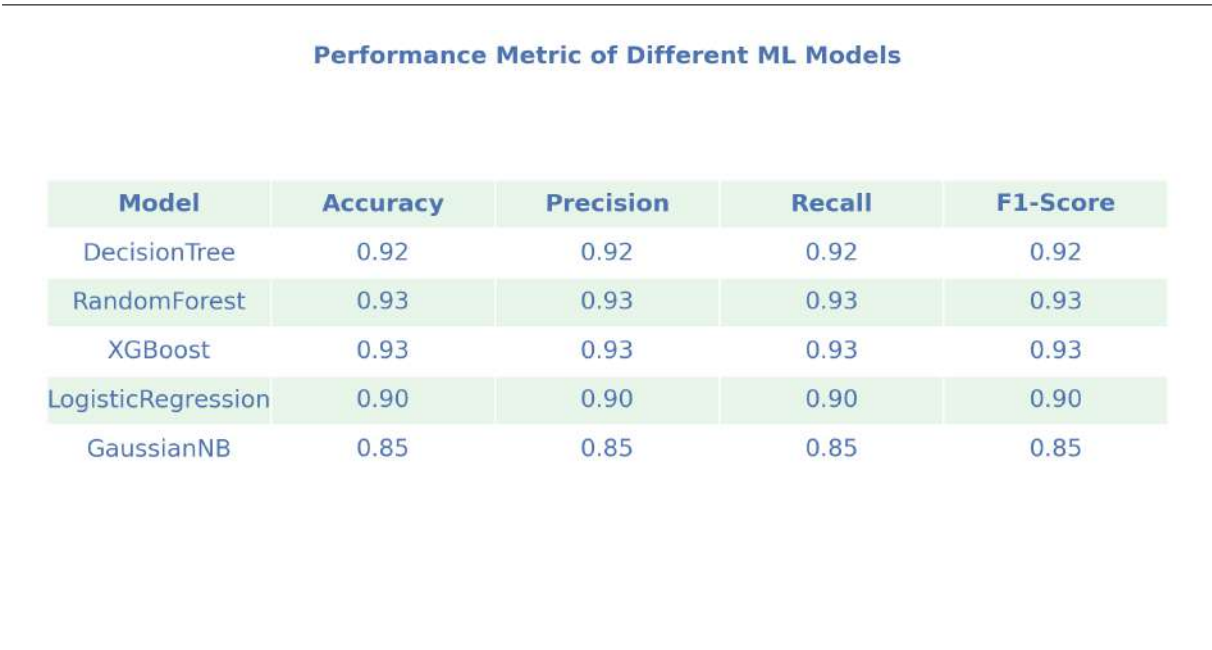


Figure 10: Comprehensive performance metrics (Precision, Recall, F1-Score) for each classification model evaluated against edge resource constraints (Sharma et al., 2021) (Source: Authors)

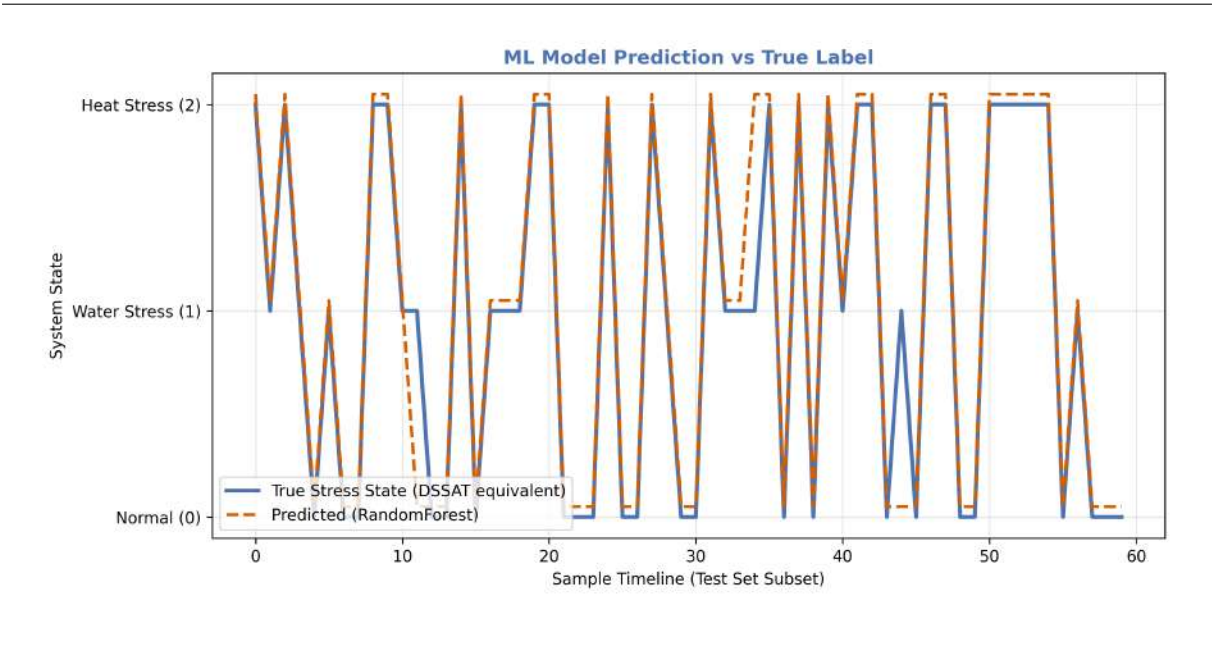


Figure 11: Predicted vs Actual class mapping validation across test samples, demonstrating alignment with ML evaluation practices in precision agriculture (Kamilaris and Prenafeta-Boldú, 2018) (Source: Authors)

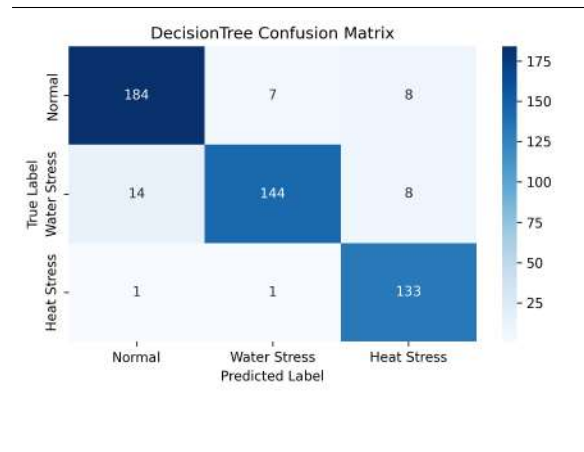


Figure 12: Decision Tree Confusion Matrix (Source: Authors)

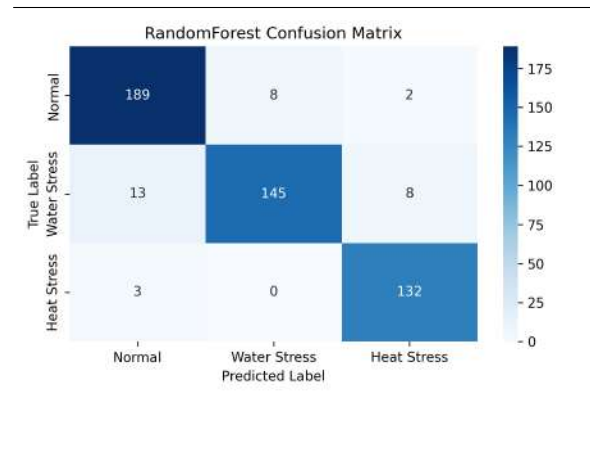


Figure 13: Random Forest Confusion Matrix (Source: Authors)

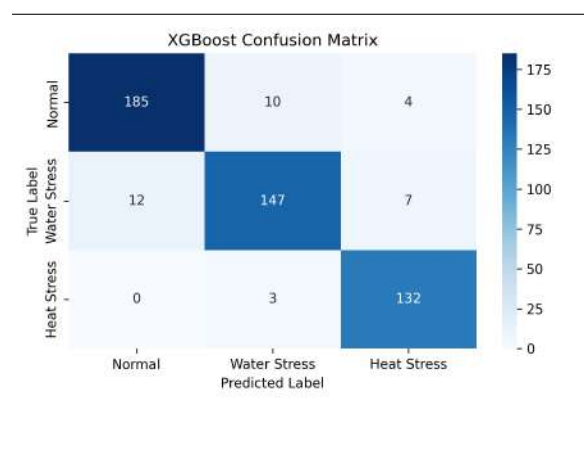


Figure 14: XGBoost Confusion Matrix (Source: Authors)

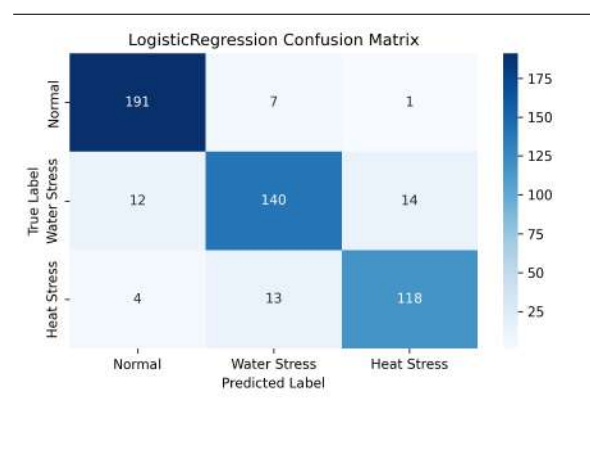


Figure 15: Logistic Regression Confusion Matrix (Source: Authors)

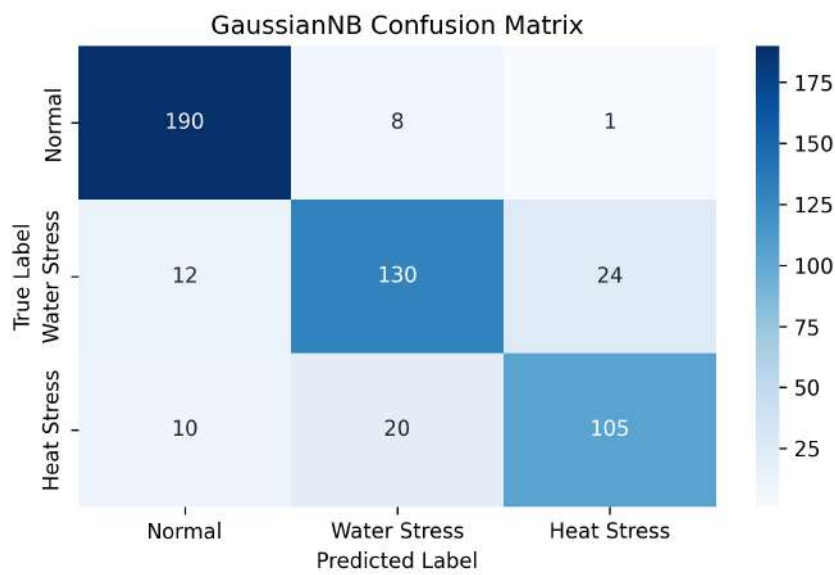


Figure 16: Gaussian Naive Bayes Confusion Matrix (Source: Authors)

Memory Footprint (Flash/RAM): Forests are the ones that have high accuracy (93.20) but are extremely large and unrolled arrays of if/else. On the other extreme, absolute minimal models such as Logistic Regression (89.80%), Gaussian Naive Bayes (85.00%), etc. require virtually zero memory overhead for their dot products.

Latency of inference: Since tree depths were held fixed (max depth 5), the deterministic decision-making process does not go beyond the RTOS deadline constraints, and as such, does not cause any starvation of high-priority analog ADC sensor loops.

Index Feature Reliance: Feature engineering allowed standard Decision Trees to achieve 92.20% which places thermal anomalies against moisture stress and demonstrates that pre-processing tasks are a huge boost to the success of TinyML.

7 Decision Logic, Outlier Rejection, and Actuation

The most serious defect of a naive automated system is so-called chatter - quick swerving of hardware actuators due to temporal environmental differences (Fernández-Ahumada et al., 2020). To illustrate, a temporary increase in the CWSI calculation caused by an impromptu and short burst of hot wind may cause an isolated ML inference of heat stress. Switching on of an irrigation pump on single inference is a waste of water and mechanical relays are degraded.

In a solution to this, the vDecisionTask uses a Temporal Sliding Window to debounce inferences of machine learning.

7.1 Temporal Hysteresis

It is a mathematical stable moving average, the system needs a firm before it takes over a mechanical tool. On the event that the ML task gives out STATE_WATER_STRESS, a counter is increased. On the next inference, in case the result of the inference is STATE_NORMAL, a counter is zeroed. The debouncing window (5 cycles, 10 seconds) is to make sure that the physical

```

// Require 10 consecutive seconds of stress before enabling the
// GPIO tool
if (water_stress_count >= DEBOUNCE_WINDOW_SIZE) {
    official_state = STATE_WATER_STRESS;
    water_stress_count = DEBOUNCE_WINDOW_SIZE; // Cap value to
    prevent overflow
    HAL_GPIO_WritePin(OUTPUT_PORT, OUTPUT_PIN, GPIO_PIN_SET);
}

```

Figure 17: Temporal hysteresis debounce logic for GPIO actuation control (Source: Authors)

device (which is mapped to the GPIO PA5) is not brought into action unless a physiological stress event has been confirmed by the long-lasting environmental pressure.

This is achieved by hysteresizing software through RTOS queues, which filter signal spikes which often afflict the agricultural IoT. The debouncer works in a similar way as the electrical switch debouncing, but on a macroscopic biological level. This greatly increases the lifetime of the various mechanical relay tools that are interconnected and maximize water and power conservation activities.

8 Telemetry, Data Collection, and System Validation

Although the system functions independently in the offline environment, continuous supervised learning and model improvement entail the derivation of quality and perfectly structured datasets.

8.1 Memory-Safe Formatting and Mutexes

Within an RTOS, when using several tasks trying to print to a single UART terminal at the same time the serial buffer will become corrupt, thus the text strings will not be readable. The firmware uses a global thread safe Mutex (`xSemaphoreTake(xMutexUART, portMAX_DELAY)`). This will ensure that logging utilities will be given time to wait, so that they can be stable with regard to terminal footprint. In addition, even native `printf` functions that are capable of working with floating points make standard executions crash on constrained boards unless linker-specifics are modified. To transmit floats safely, the system scales the floats to integers (e.g. `(int)(procData.cwsi * 100)`) to be transferred safely by the serial system.

8.2 The Dedicated CSV Stream

The B1 switch breaks the normal stream of logs and transforms the communication stream into a native CSV exporter when it is toggled. It strips of all RTOS metadata (timestamps, debug tags) and streams delimited integers: `Moisture_Norm`, `Temp_Norm`, `Hum_Norm`, `Light_Norm`, `CWSI`, `STI`, `SSPI`. This structural streaming of a naked CSV permits a connected field laptop to consume thousands of rows of well-formed aggregate of dataset directly into Python/Pandas without going through a post-processing stage of cleaning up the string.

Robust diagnostic output strategies are what validate the edge logic models in the field. Researchers need to have pure arrays when making large agricultural matrices. The seamless

physical switch integration means a field technician requires zero programming knowledge; a single button press is a seamless transition of the node between a defensive actuator and a passive, highly granular data sink. The FreeRTOS provides a high level of accuracy in the timing of the rows in the CSV data which is required in future time-series model training.

9 Conclusion

This work has been able to show that it is feasible to marry the precision agricultural theorems with the strictly constrained C-based computation and RTOS determinism. The system is a wholly independent edge node, meaning it will avoid systemic susceptibilities of IoT networks, including rural connectivity loss, and server latency. By applying advanced agronomic indices (CWSI, STI, SSPI), lightweight deterministic ML classification, and effective temporal debouncing, the hardware is capable of measuring, validating, and physically mitigate crop stress in real time.

Migrating from cloud-reliant systems to edge inference at the network periphery is not only a technical optimization, but an essential requirement of developing countries and remote rural farms where internet connectivity is completely unreliable. Using the ability to have the Nucleo make complex mathematically sound irrigation decisions, agriculture yields can be automatically defended against the unexpected climatic shocks, heatwaves, and drought conditions.

Future developments will focus on bridging the precise I2C physical sensor mapping directly into live-field, large-scale tomato crop rows. Furthermore, utilizing the node's standalone CSV logging capability to extract seasonal field data will allow future researchers to continuously fine-tune the offline machine learning models, adapting them to highly specific regional micro-climates and maximizing long-term crop survivability with minimal ecological impact.

10 Acknowledgments

The author extends gratitude to the developers of FreeRTOS and the STMicroelectronics hardware community for providing the fundamental embedded tools that made this deterministic architecture possible. Acknowledgment is also given to the agronomic researchers whose empirical data on *Solanum lycopersicum* physiology allowed for the accurate calibration of the embedded stress indices.

11 Where to look for further information

For further information regarding the foundational principles of edge computing, TinyML, and IoT architectures in agriculture, academic journals such as *Computers and Electronics in Agriculture*, *IEEE Internet of Things Journal*, and *Precision Agriculture* are invaluable resources. They frequently publish peer-reviewed implementation models detailing the convergence of hardware and agronomy.

For hands-on hardware development, detailed documentation on hardware-level real-time operating systems can be found in the official FreeRTOS reference manuals (freertos.org). Developers seeking to replicate the MCU infrastructure should heavily consult STMicroelectronics' *STM32 Cortex-M4* technical training libraries and the comprehensive *STM32Cube HAL (Hardware Abstraction Layer)* driver manuals.

For specialized insights into the complex translation of machine learning models into bare-metal C code, developers should explore the official documentation for the `micromlgen` and

emlearn Python libraries on GitHub. These repositories provide vital, open-source implementation pipelines for compressing Random Forests, Support Vector Machines, and Decision Trees into standard C-header arrays perfectly compatible with resource-constrained microcontrollers.

References

- Ayaz, M., Ammad-Uddin, M., Sharif, Z., Mansour, A., & Aggoune, E.-H. M. (2019). Internet-of-Things (IoT)-Based Smart Agriculture: Toward Making the Fields Talk. *IEEE Access*, 7, 129551–129583. <https://doi.org/10.1109/ACCESS.2019.2932609>
- Fernández-Ahumada, A. et al. (2020). IoT-Based Smart Irrigation Systems: An Overview. *Sensors*, 20(4). <https://doi.org/10.3390/s20041043>
- Hiremath, S. et al. (2025). Smart Farming System for Soil-Centric Crop Selection and Fertilizer Content Optimization Using IoT and Ensemble Learning. *Proc. IEEE ICWITE*. <https://doi.org/10.1109/ICWITE64848.2025.11306954>
- Jawad, A. et al. (2017). Energy-Efficient Wireless Sensor Networks for Precision Agriculture. *Sensors*, 17(8). <https://doi.org/10.3390/s17081783>
- Jayaraman, P. P. et al. (2017). Internet of Things Platform for Smart Farming. *IEEE Internet of Things Journal*, 4(6), 1819–1829. <https://doi.org/10.1109/JIOT.2017.2737993>
- Kamilaris, A., Kartakoullis, A., & Prenafeta-Boldú, F. X. (2017). A Review on the Practice of Big Data Analysis in Agriculture. *Computers and Electronics in Agriculture*, 143, 23–37. <https://doi.org/10.1016/j.compag.2017.09.037>
- Kamilaris, A. & Prenafeta-Boldú, F. X. (2018). Deep Learning in Agriculture: A Survey. *Computers and Electronics in Agriculture*, 147, 70–90. <https://doi.org/10.1016/j.compag.2018.02.016>
- Kamilaris, A. & Prenafeta-Boldú, F. X. (2019). Transfer Learning in Agriculture. *Computers and Electronics in Agriculture*, 156, 547–554. <https://doi.org/10.1016/j.compag.2018.12.020>
- Khaliq, R. et al. (2023). AI-Driven Smart Agriculture: An Integrated Approach for Soil Analysis, Irrigation and Crop-Fertilizer Recommendations. *Sustainability*, 15(6). <https://doi.org/10.3390/su15064821>
- Khanna, A. & Kaur, S. (2020). Internet of Things (IoT), Applications and Challenges: A Comprehensive Review. *Wireless Personal Communications*, 114(2), 1687–1762. <https://doi.org/10.1007/s11277-020-07446-4>
- Li, R. et al. (2019). Edge Computing for Smart Agriculture. *Future Generation Computer Systems*, 97, 369–379. <https://doi.org/10.1016/j.future.2019.02.027>
- Liakos, R. et al. (2018). Machine Learning in Agriculture: A Review. *Sensors*, 18(8). <https://doi.org/10.3390/s18082674>
- Manogaran, M., Alazab, M., Muhammad, K., & de Albuquerque, V. H. C. (2021). Smart Sensing Based Functional Control for Reducing Uncertainties in Agricultural Farm Data Analysis. *IEEE Sensors Journal*, 21(16), 17469–17484. <https://doi.org/10.1109/JSEN.2021.3054561>

- Movla, M. N., Abawajy, J., & Chowdhury, M. (2020). Internet of Things and Wireless Sensor Networks for Smart Agriculture: A Survey. *IEEE Access*, 8, 123659–123689. <https://doi.org/10.1109/ACCESS.2020.3007965>
- Navarro-Hellín, H. et al. (2019). A Decision Support System for Managing Irrigation in Agriculture. *Computers and Electronics in Agriculture*, 162, 537–550. <https://doi.org/10.1016/j.compag.2019.05.030>
- Navarro, H. et al. (2021). IoT-Based Smart Sensor Systems for Agriculture. *Sensors*, 21(6). <https://doi.org/10.3390/s21062137>
- Pathak, A., Brown, M., & Best, S. (2019). A Systematic Review of IoT in Agriculture. *Precision Agriculture*, 20, 873–906. <https://doi.org/10.1007/s11119-019-09608-8>
- Penmetsa, Y. K. et al. (2021). Smart Precision Agriculture Using Deep Neural Networks. *IEEE Access*, 9, 102512–102530. <https://doi.org/10.1109/ACCESS.2021.3098034>
- Rupanagudi, S. R. et al. (2015). A Novel Cloud Computing Based Smart Farming System. *IEEE Int. Conf. Computational Intelligence and Computing Research*. <https://doi.org/10.1109/ICCIC.2015.7435786>
- Sharma, A., Jain, A., Gupta, P., & Chowdary, V. (2021). Machine Learning Applications for Precision Agriculture: A Comprehensive Review. *IEEE Access*, 9, 4843–4873. <https://doi.org/10.1109/ACCESS.2020.3048415>
- The Hindu. (2014). Tomato price dips to below Re 1/kg; farmers in despair. *The Hindu*. Retrieved from <https://www.thehindu.com/news/national/karnataka/tomato-price-dips-to-below-re-1-kg-farmers-in-despair/article5591061.ece>
- Vasisht, S. T. et al. (2017). FarmBeats: An IoT Platform for Data-Driven Agriculture. *Proc. USENIX NSDI*. <https://doi.org/10.5555/3154630.3154638>
- Wolfert, J., Ge, L., Verdouw, C., & Bogaardt, M.-J. (2017). Big Data in Smart Farming – A Review. *Agricultural Systems*, 153, 69–80. <https://doi.org/10.1016/j.agry.2017.01.023>