System Specification Document (SSD)

Machine Learning Operations

Cartago · Tonet · Toso · Zambonini

Document Status. The following document presents a preliminary and evolving version of the system specification. The described requirements, design choices, and architectural decisions may be subject to modification as the project advances. Accordingly, this document should be regarded as a working draft rather than a finalized specification.

Contents

1 Problem Definition

Effective plant cultivation requires consistent monitoring of environmental conditions to maintain optimal growth and detect potential issues early. Manual observation and periodic measurements provide only snapshots of environmental data, lacking the continuity necessary for comprehensive analysis.

Without continuous and reliable data collection, gradual environmental changes and anomalous patterns often go undetected until they have already impacted plant health. This reactive approach limits the ability to intervene proactively and maintain ideal growing conditions. This project aims to develop an automated environmental monitoring system that collects, processes, and analyzes sensor data.

The system will provide insights into environmental conditions and enable early detection of anomalous behavior, facilitating timely intervention and optimized plant care.

1.1 Business Problem

The goal of this project is to design a smart plant monitoring system capable of collecting, aggregating and analyzing environmental data using sensors and machine learning techniques. The system aims to support **data-driven decision making**, detect drifts, and provide real-time visualization of plant conditions.

1.2 ML Formulation

From a Machine Learning perspective, the problem can be formulated as a **time-series analysis and monitoring**.
The Machine Learning model will consider the following tasks:

- Statistical aggregation of sensor data;
- Detection of data drift and anomalous trends;
- Predictive modeling of plant watering conditions.

The Machine Learning model will consider the following input features (sensors):

- Light intensity;
- Temperature;
- Humidity;
- Soil moisture.

1.3 KPIs

The global effectiveness of the system will be evaluated using the following KPIs.

- **Data Availability**: number of successfully stored sensor data;
- **Data Latency**: delay between data acquisition and cloud availability;
- **System Reliability**: stability of sensor readings over time;

- **Drift Detection Accuracy**: ability to identify and plot significant changes in data distributions;
- **Dashboard Responsiveness**: load and refresh time of the visualization interface.

2 Data Specification

2.1 Data Sources

Data are collected from **four environmental sensors** attached to a plant:

- Light Sensor;
- Temperature Sensor;
- Humidity Sensor;
- Soil Moisture Sensor.

Each sensor is connected to an Arduino microcontroller interface, which acts as the data acquisition unit. Table 1 summarizes the measurement ranges and units associated with each sensor.

Table 1: Sensor measurement ranges and scales

Sensor	**Measurement Range**	**Unit / Scale**
Humidity	0 – 100	Percentage (%)
Temperature	0 – 100	Degrees Celsius (°C)
Light	0 – 1023	ADC scale (0 = dark, 1023 = max light)
Soil Moisture	0 – 1023	ADC scale (0 = dry soil, 1023 = fully wet)

Light intensity and soil moisture readings are expressed on the native analog-to-digital converter (ADC) scale of the sensors and should therefore be interpreted as **qualitative indicators** of environmental conditions rather than absolute physical measurements. For soil moisture, a reference threshold value of approximately 840 is expected under optimal watering conditions.

2.2 Data Flow

The data flow describes the end-to-end pipeline through which sensor measurements are acquired, processed, stored, and made available for analysis and visualization. Data are progressively transformed through a sequence of processing stages designed to ensure reliability, temporal consistency, and accessibility. This structured flow enables continuous data acquisition, aggregation over fixed time windows, and centralized cloud logging, while also supporting secure access and real-time monitoring through dedicated visualization tools

1. Sensors sample raw signals every 10 seconds, which are summarized through a 1-minute temporal window;
2. The Arduino board computes local statistics and transmits feature-based data every 5 minutes;

3. A weighted temporal aggregation over the last 15 minutes is applied to the sensor features during the Arduino-to-PC data transfer;
4. Processed sensor data are logged to the cloud using **Weights and Biases** (W&B);
5. Authorized clients access the data through secure API keys;
6. Data are visualized and continuously monitored via a **Streamlit Dashboard**.

2.3 Quality

Data quality is essential to ensure that the collected measurements accurately represent the environmental conditions of the plant. Basic checks are applied to identify potential issues that may affect the reliability of the data.
To ensure data reliability, the following aspects are considered:

- Handling missing or corrupted sensor readings;
- Noise reduction through temporal aggregation;
- Detection of out-of-range values;
- Timestamp consistency across sensors.

2.4 Pre-processing

Preprocessing steps include:

- Timestamp alignment;
- Weighted averaging over fixed time windows;
- Normalization of sensor values;
- Formatting for cloud logging and visualization

3 Functional Requirements

FR01	The Arduino Microcontroller collects environmental data from four sensors measuring light, temperature, humidity, and soil moisture.
FR02	Sensor measurements are sampled every 10 seconds for each sensor.
FR03	Sensor data are summarized using a 1-minute temporal window to compute statistical features.
FR04	A representative set of features is generated and transmitted every 5 minutes for each sensor.
FR05	The Arduino Microcontroller forwards collected data to a local server for processing.
FR06	The Arduino performs local processing of sensor data and transmits the resulting features to the local server.
FR07	Aggregated sensor data is uploaded to a cloud-based platform (W&B) for long-term storage.
FR08	The data pipeline integrates with an experiment tracking platform to log time-series measurements.
FR09	Access to cloud-stored data is controlled through API keys assigned to authorized users.
FR10	Authorized clients can retrieve and upload sensor data through secure interfaces.
FR11	Visualization of sensor data is provided through a web-based dashboard.
FR12	Historical sensor data can be explored through the dashboard for trend analysis.
FR13	Changes in data distributions are monitored to identify potential data drift.
FR14	Abnormal patterns or drift events are signaled to the user interface.
FR15	System operations and detected anomalies are recorded for monitoring and debugging purposes.

4 Non Functional Requirements

NFR01	The system is designed to operate continuously under normal environmental and network conditions.
NFR02	Data processing and cloud logging occur with minimal delay after acquisition.
NFR03	The system architecture supports scalability for additional sensors or monitored plants.
NFR04	Secure communication mechanisms are used for data exchange between system components.
NFR05	Stored data remains protected against unauthorized modification or loss.
NFR06	Temporary network or power failures do not permanently compromise data availability.
NFR07	Software components follow a modular structure to simplify maintenance and updates.
NFR08	The system is designed to be deployable across different operating systems without functional degradation.
NFR09	The user interface prioritizes clarity and ease of use for effective data interpretation.
NFR10	Logging and monitoring capabilities facilitate performance evaluation and troubleshooting.

5 Project Architecture

5.1 Training Phase

5.2 Validation Phase

5.3 Deployment phase

5.4 Monitoring phase

6 Risk Analysis

Risk	Description
Sensor Failure	One or more sensors may malfunction, degrade over time, or provide inaccurate measurements, leading to unreliable data collection.
Data Loss	Interruptions in communication, power outages, or hardware issues may cause partial or complete loss of sensor data.
Noise in Data	Environmental disturbances or sensor limitations may introduce noise and fluctuations in the collected measurements.
Security Breach	Unauthorized access to the cloud platform or APIs may compromise data integrity and confidentiality.
Scalability Issues	The system may face performance or management challenges when integrating additional sensors or monitored plants.