**1. Abstract / Executive Summary**

This project, titled "Environmental Classification for Plants", presents an intelligent IoT-based system that leverages real-time sensor data and machine learning for classifying environmental conditions critical to plant health. Utilizing an ESP32-S3 microcontroller and sensors such as DHT11, MQ135, LDR, and a soil moisture sensor, the system collects environmental data, transmits it over MQTT, and processes it using a trained machine learning model. The prediction results are logged into both InfluxDB for local visualization and ThingSpeak for cloud access. The ultimate aim is to enhance decision-making in precision agriculture and greenhouse monitoring systems by automating environmental assessment.

**2. Table of Contents**

* Abstract / Executive Summary
* Table of Contents
* Introduction
* Literature Review
* Methodology / System Design
* Implementation
* Results & Discussion
* Testing & Validation / Limitations
* Conclusion & Future Work
* References
* Links

**3. Introduction**

**Background & Motivation**

With the growing need for sustainable agriculture, environmental monitoring is more crucial than ever. Traditional systems require manual intervention and often lack precision. By using IoT devices and artificial intelligence, we can monitor environmental factors more accurately and make data-driven decisions to support healthy plant growth.

**Problem Statement**

Farmers often face challenges due to unpredictable environmental changes, leading to reduced crop yields. Manual monitoring is not scalable. This project addresses the gap by automating environmental monitoring and classification using low-cost IoT devices and machine learning.

**Project Goals**

* Collect real-time environmental sensor data using ESP32.
* Train a machine learning model to classify the environmental condition.
* Visualize the predictions and sensor values in real time using InfluxDB and ThingSpeak.
* Enable the system to support smart agricultural decision-making.

**4. Literature Review**

Recent research has shown the potential of integrating IoT with agriculture to improve crop management. Microcontrollers like ESP32 offer built-in Wi-Fi and GPIOs for sensor integration. Sensors like DHT11 (for temperature and humidity), MQ135 (for air quality), LDR (for light intensity), and soil moisture sensors have proven effective in monitoring agricultural environments.

MQTT is a lightweight protocol ideal for IoT data transmission due to its low overhead. InfluxDB is a powerful time-series database for storing real-time data, while ThingSpeak provides a user-friendly cloud platform for data visualization. Machine learning models, especially tree-based classifiers like Random Forests, offer excellent performance for multi-class classification problems.

**5. Methodology / System Design**

**5.1 Hardware Components**

**ESP32-S3 Dev Module:** Acts as the central microcontroller collecting and sending sensor data.

**DHT11:** Measures ambient temperature and humidity.

**MQ135:** Detects air quality by sensing gases like CO2 and NH3.

**LDR:** Detects sunlight intensity.

**Soil Moisture Sensor:** Determines the moisture content in the soil.

**Wiring Summary:**

DHT11 → GPIO 4

MQ135 AO → GPIO 15, DO → GPIO 14

LDR AO → GPIO 18, DO → GPIO 12

Soil AO → GPIO 16, DO → GPIO 5

**5.2 Software Architecture**

**ESP32:** Reads analog and digital values from sensors and publishes them via MQTT.

**Python Backend:** Subscribes to the MQTT topic, parses sensor data, predicts the environment class using a Random Forest model, and logs the results into InfluxDB and ThingSpeak.

**Model Features:** temperature, humidity, mq135\_ao, ldr\_ao, soil\_ao

**Classes Predicted:** Optimal, Dry, Moist, Wet, LowLight, HighPollution

**Architecture Flow:**

Sensor → ESP32 → MQTT → Python Script → ML Prediction → Store in InfluxDB/ThingSpeak

**6. Implementation**

**Hardware Setup**

All sensors were mounted on a breadboard and connected to the ESP32-S3 module using jumper wires. The ESP32 was powered via USB. The analog outputs give precise readings, while digital outputs are used for threshold-based detection.

**Firmware and Backend**

ESP32 was programmed in Arduino IDE to read sensor values and publish a JSON message every 5 seconds over MQTT.

A Python script subscribed to the agri/sensor/data topic and parsed the JSON.

Using a pre-trained Random Forest model, predictions were made and saved to InfluxDB and ThingSpeak.

**Issues Faced and Resolved**

MQTT rc=-2 error: Resolved by ensuring Mosquitto was bound to the correct LAN IP and allowed through the firewall.

InfluxDB error ("token required"): Fixed by assigning proper tokens and using synchronous write API.

Unsupported input type in InfluxDB: predicted\_class was moved from a field to a tag to avoid aggregation issues.

**7. Results & Discussion**

**Real-Time Data Collection**

Sensor readings such as temperature, humidity, air quality, and soil moisture were successfully collected and transmitted.

**Machine Learning Performance**

The Random Forest model trained on synthetic data performed with 100% accuracy on the test set. The classes were well-separated in the feature space, which contributed to the high precision.

**Visualization**

**InfluxDB:** Time-series plots showed changes in environmental values and predicted classes.

**ThingSpeak:** Graphs updated every 15 seconds to show real-time trends.

**8. Testing & Validation / Limitations**

**Testing**

Simulated each environmental class by changing sensor inputs.

Verified that predictions matched expected class labels.

**Limitations**

Synthetic data used for training might not reflect sensor noise in real environments.

ThingSpeak’s free tier limits updates to one every 15 seconds.

Only 8 fields are available per channel on ThingSpeak.

**9. Conclusion & Future Work**

This project demonstrates the feasibility of an end-to-end IoT system that classifies environmental conditions using live sensor data and machine learning. The combination of MQTT, InfluxDB, and ThingSpeak allows for real-time monitoring and cloud-based visualization.

**Future Work**

Collect real labeled data under diverse environmental conditions to improve model generalization.

Expand the number of sensors (e.g., CO2, soil pH, UV).

Create a user-facing dashboard (mobile or web) for live decision-making.

Add automated actions (e.g., watering, ventilation) based on classification.

**10. References**

MQTT.org: [https://mqtt.org](https://mqtt.org/)

ThingSpeak Docs: <https://thingspeak.com/docs>

InfluxDB Documentation: [https://docs.influxdata.com](https://docs.influxdata.com/)

Arduino & ESP32 Libraries

Scikit-learn Random Forest Classifier

**11. Links**

GitHub Repo: **https://github.com/hamza01012/Lab-work/tree/main/Project\_\_\_\_\_\_Environmental%20Classification%20for%20Plants**