Shivam Giri
2021UEA4538
ECAM 1
18 November 2024

Smart Irrigation System - AquaSmartAgri

Aim:

The aim of this project is to develop a smart irrigation system that leverages IoT and AI to optimize water usage in agriculture. The system collects data such as temperature and humidity, processes it through an ML model, and provides real-time irrigation suggestions to farmers. This helps in conserving water and improving agricultural productivity.

Material Used:

- **Virtual Hardware Platform**: Wokwi (to simulate sensors)
- **IoT Sensors**:
  - **Temperature and Humidity Sensor** (virtual)
  - **Visual sensors for pest and weed data** (conceptual)(future scope)
- **Cloud Platform**: ThingSpeak for real-time data visualization and storage
- **Programming Environment**:
  - Python for model training and backend processing
- **Output Device**: Web interface for displaying predictions

Theory:

This project integrates **IoT technology** and **machine learning** to create a data-driven irrigation system. The IoT sensors capture environmental data (temperature, humidity, pest/weed visuals), which is then transmitted to ThingSpeak for visualization. The data is subsequently fed into a **Random Forest** machine learning model that analyzes it and provides irrigation suggestions.

Random Forest is an ensemble learning method that constructs multiple decision trees and combines their outputs for more reliable predictions. It is suitable for this project due to its robustness and high accuracy in handling complex datasets.

Software Used:

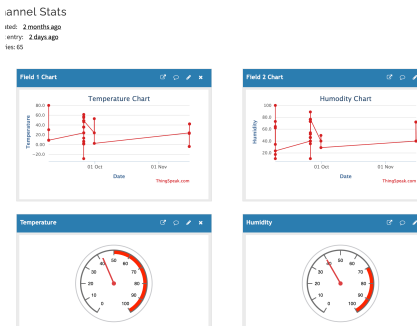
- **Python Libraries**:
  - `pandas` for data handling
  - `scikit-learn` for model training
  - `time` for delay functionality
  - `IPython.display` for displaying HTML output
- **Web Technologies**:
  - HTML and CSS for user interface design

- **Platforms**:
  - **Wokwi** for virtual hardware simulation
  - **ThingSpeak** for cloud data storage and visualization
- **Streamlit** or **Flask** for creating a web interface

Code Section:

```
//channel api key : NIDI5DG5WLF3YYWN

#include <WiFi.h>
#include "DHTesp.h"
#include "ThingSpeak.h"

const int DHT_PIN = 15;
const int LED_PIN = 13;
const char* WIFI_NAME = "Wokwi-GUEST";
const char* WIFI_PASSWORD = "";
const int myChannelNumber =2648719 ;
const char* myApiKey = "NIDI5DG5WLF3YYWN";
const char* server = "api.thingspeak.com";

DHTesp dhtSensor;
WiFiClient client;

void setup() {
  Serial.begin(115200);
  dhtSensor.setup(DHT_PIN, DHTesp::DHT22);
  pinMode(LED_PIN, OUTPUT);
  WiFi.begin(WIFI_NAME, WIFI_PASSWORD);
  while (WiFi.status() != WL_CONNECTED){
    delay(1000);
    Serial.println("Wifi not connected");
  }
  Serial.println("Wifi connected !");
  TempAndHumidity  data = dhtSensor.getTempAndHumidity();
  ThingSpeak.setField(1,data.temperature);
  ThingSpeak.setField(2,data.humidity);
  if ((data.temperature > 35 || data.temperature < 12
  || data.humidity > 70 || data.humidity < 40)) {
    digitalWrite(LED_PIN, HIGH);
  }else{
    digitalWrite(LED_PIN, LOW);
  }

  int x = ThingSpeak.writeFields(myChannelNumber,myApiKey);

  Serial.println("Temp: " + String(data.temperature, 2) + "°C");
  Serial.println("Humidity: " + String(data.humidity, 1) + "%");

  if(x == 200){
    Serial.println("Data pushed successfull");
  }else{
    Serial.println("Push error" + String(x));
  }
  Serial.println("---");

  delay(10000);
}
```

ThingSpeak Output:



Google Colab link - https://colab.research.google.com/drive/1PGcT8LMeE3msYe0KlH3YjYvlHozU_NEK?usp=sharing

User Interface for Farmers: Simple UI/UX