Plant Disease Detection Using ML & IOT

Ganesh Patil [1], Swati Bharati[2], Shubham Mane[3], Gajanan Ganakhedkar[4], Prof. Priyanka. S. Patil[5]
[1,2,3,4]*Undergrad. Student, Dept. of Information Technologies SKN Sinhgad Institute of Technology & Science, Lonavala, Maharashtra*
[5]*Asst.Professor, Dept. Of Information Tech, SKN Sinhgad Institute of Technology & Science, Lonavala, Maharashtra*

Abstract: Plant diseases can cause significant crop losses, leading to economic and food security challenges. Early detection and treatment of plant diseases are essential to minimize losses. IoT and machine learning (ML) can be used to develop a plant disease prediction system that can monitor crop health and send alerts to farmers when diseases are detected. This abstract proposes a plant disease prediction system using IoT and ML using ESP32, temperature sensor, humidity sensor, sun intensity sensor, moisture sensor, solar panel, battery, and Blynk app. The system consists of the following components:
ESP32 microcontroller: The ESP32 is a low-cost, low-power microcontroller that is ideal for IoT applications. It has built-in Wi-Fi and Bluetooth connectivity, which makes it easy to connect to the cloud and send data to farmers. Temperature sensor: The temperature sensor measures the ambient temperature around the plant. Humidity sensor: The humidity sensor measures the relative humidity around the plant. Sun intensity sensor: The sun intensity sensor measures the amount of sunlight that the plant is receiving. Moisture sensor: The moisture sensor measures the moisture content in the soil around the plant. Solar panel: The solar panel provides power to the ESP32 and sensors. Battery: The battery provides backup power in case the solar panel is not able to generate enough power. Blynk app: The Blynk app is a mobile app that can be used to monitor the data from the sensors and receive alerts when diseases are detected.

Keywords: ESP32, Temperature sensor, Humidity sensor, Sun intensity sensor, Moisture sensor, Solar panel, Battery, Blynk app, Plant disease prediction, IoT, ML.

INTRODUCTION

A plant disease prediction system is a computer-based system that uses machine learning to predict the occurrence of plant diseases. These systems can be used to help farmers and other agricultural professionals make informed decisions about disease prevention and treatment. Plant disease prediction systems typically work by analysing data about the plant, the environment, and the disease itself. This data can include information about the plant species, the growth stage, the weather conditions, and the symptoms of the disease. The system then uses this data to train a machine learning model, which is a mathematical algorithm that can learn to identify patterns in the data. Once the model is trained, it can be used to predict the likelihood of a plant developing a particular disease. This information can be used by farmers to make decisions about when to spray pesticides, when to harvest crops, and other important management practices. Plant disease prediction systems can also be used to forecast the spread of diseases. This information can be used by governments and other organizations to develop policies and programs to prevent and control disease outbreaks. Plant diseases can cause significant losses in crop yields and quality. Early detection and treatment of plant diseases is essential to minimize these losses. Traditional methods of plant disease detection, such as visual inspection by farmers, can be time-consuming and inaccurate. Internet of Things (IoT) and machine learning (ML) technologies can be used to develop more efficient and accurate plant disease detection systems. IoT devices can be used to collect data on various environmental factors, such as temperature, humidity, sun intensity, and soil moisture. This data can then be fed into ML models to predict the likelihood of plant diseases occurring.

The following system architecture can be used to develop a plant disease prediction system using IoT and ML:

•ESP32 microcontroller: The ESP32 is a low cost and low-power microcontroller that can be used to develop IoT devices. •Environmental sensors: Temperature, humidity, sun intensity, and moisture sensors can be used to collect data on the environmental conditions in the crop field. •Solar panel and battery: The solar panel can be used to power the ESP32 microcontroller and the environmental sensors. The battery can be used to store power for use during nighttime or when the solar panel is not receiving enough sunlight.

•Blynk app: The Blynk app can be used to monitor the data collected by the environmental sensors and to receive alerts when plant diseases are predicted.

MOTIVATION

Plant diseases can cause significant losses in crop yield and quality, leading to economic hardship for farmers and food insecurity for consumers. Early detection and treatment of plant diseases are essential to minimize these losses. However, traditional methods of plant disease detection and monitoring are often time-consuming and labour-intensive, and they may not be effective in detecting diseases at their early stages. IoT- and ML-based plant disease prediction systems can address these challenges by providing real-time monitoring of plant health and early detection of diseases. These systems can collect data from a variety of sensors, such as temperature sensors, humidity sensors, sun intensity sensors, and moisture sensors. This data can then be processed using ML algorithms to identify patterns that are indicative of plant dis-eases. Once a disease is detected, the system can send alerts to farmers so that they can take timely action to prevent the spread of the disease Early detection of plant diseases is crucial for effective management. However, traditional methods of disease detection are often labour-intensive and time consuming. IoT and ML-based systems can automate the process of disease detection and prediction, allowing farmers to identify and respond to diseases early on. IoT and ML-based systems can also help farmers to better understand the environmental factors that contribute to plant diseases. This information can be used to develop targeted disease prevention and management strategies. IoT and ML-based systems can be used to monitor crop health remotely. This can be particularly beneficial for farmers who have large or remote fields.

OBJECTIVE

Early detection of plant diseases: By collecting data on environmental conditions and plant health, IoT and ML models can detect plant diseases at an early stage, before they cause significant damage. Improved crop yields: Early detection and treatment of plant diseases can help to improve crop yields and reduce the need for pesticides and other chemicals. Reduced environmental impact: Reducing the use of pesticides and other chemicals has a number of environmental benefits, including improved water quality and soil health. Increased farmer profitability: By improving crop yields and reducing costs, IoT and ML based plant disease prediction systems can help farmers to increase their profitability. Detect plant diseases early and accurately. This can be done by using machine learning to train a model on a dataset of images of healthy and diseased plants. The model can then be used to classify new images of plants and predict whether they are healthy or diseased. Provide farmers with real-time information about the health of their crops. This can be done by using IoT to collect data from the sensors and send it to a cloud server or mobile app. Farmers can then use this information to make informed decisions about their crops, such as when to water or fertilize them, or when to apply pesticides. Reduce the use of pesticides and other chemicals. By detecting plant diseases early, farmers can take preventive measures to avoid the spread of disease and the need to use chemicals. This can lead to a more sustainable and environmentally friendly agricultural sector.

A. CONTRIBUTION OF THE ANALYSIS:

The study proposed a model that identifies the critical environmental parameters considered for the development rate of a pest and suggested pest prediction on the ecological conditions. The unique feature of the study is that the model is flexible sufficiently to be involved in any pest prediction on any crop. The analysis puts forward a pest prediction model established on the unaffected sensing of environmental requirements. The proposed model stands out due to its distinctiveness regarding multiple parameters during pre-dictions and its utilization of directly sensed ecological circumstances.

PROJECT REQUIREMENT SPECIFICATION

- **Hardware Requirements**

•ESP32 microcontroller board

•Temperature sensor (e.g., DHT11, DS18B20)

•Humidity sensor (e.g., DHT11, DHT22)

•Sun intensity sensor (e.g., TSL2561)

•Moisture sensor (e.g., FC-28, YL-69)

•Solar panel (optional to power the system)

•Battery (optional to store power for the system)
•Jump wire •Other components (depending on the specific design of the system, such as resistors, capacitors, and transistors)
•Blynk app (for monitoring data and sending alerts

- **Example system design** the following is an example of a system design for plant disease prediction using IoT and ML using ESP32:

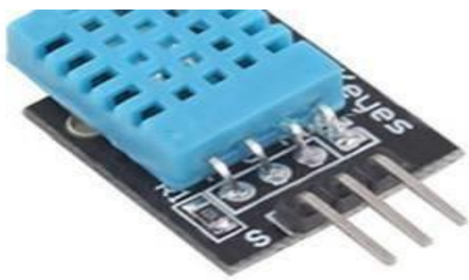
 •The ESP32 microcontroller board is the heart of the system. It is responsible for collecting data from the sensors, processing the data, and sending the data to the cloud.
 •The temperature sensor, humidity sensor, sun intensity sensor, and moisture sensor are used to collect environmental data.
 •The camera is used to capture images of the plant leaves, which can be used for image-based disease detection.
 •The solar panel and battery are used to power the system.
 •The Blynk app is used to monitor data and send alerts.

- **How the system works**
 The system works as follows:
 1.The ESP32 microcontroller board collects data from the sensors.
 2.The ESP32 microcontroller board processes the data to detect plant diseases. 3.The ESP32 microcontroller board sends the data to the cloud.
 4.The Blynk app is used to monitor the data and send alerts if plant diseases are detected.

- **ESP32** The ESP32 is a low-cost, low power system-on-a-chip (SoC) microcontroller with integrated Wi-Fi and Bluetooth. It is a popular choice for IoT and DIY projects due to its versatility and affordability.

❖ **Specifications:**
 •Dual-core 32-bit Tensilica Xtensa LX6 microprocessor
 •240 MHz clock speed
 •Up to 4 MB of flash memory
 •Up to 520 KB of SRAM
 •Integrated Wi-Fi and Bluetooth
 •Ultra-low power consumption
 •Wide range of peripherals and interfaces

- **DHT11 Temperature and Humidity Sensor** the DHT11 is a basic, low-cost temperature and humidity sensor. It is a popular choice for hobbyists and beginners due to its ease of use and affordability.

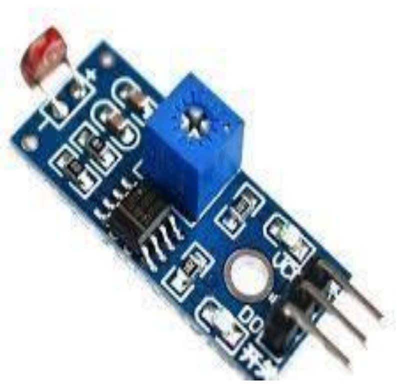


❖ **Specifications:**
 •Temperature range: 0-50°C
 •Humidity range: 20-90% RH

•Accuracy: ±2% RH and ±1°C

•Output: Digital signal (I2C)

- **LDR Sensor** An LDR sensor, or light dependent resistor, is a device whose electrical resistance changes with the amount of light incident upon it. LDR sensors are often used to detect the presence or absence of light, or to measure the intensity of light.



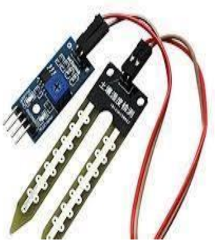
❖ **Specifications:**

•Resistance range: 10 Ω - 1 MΩ

•Peak wavelength: 540 nm

•Operating temperature range: -40°C to 100°C

- **Moisture Sensor** A moisture sensor is a device that measures the amount of water in a substance. Moisture sensors are often used in agriculture, irrigation, and weather monitoring systems.



❖ **Specifications:**

•Operating voltage: 5 V

•Operating temperature range: -10°C to 60°C

•Output: Analog signal

- **Solar Panel** A solar panel is a device that converts sunlight into electricity. Solar panels are often used to power homes, businesses, and other devices in a sustainable and environmentally friendly way.

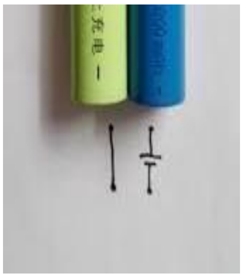
❖ **Specifications:**

•Power output: Varies depending on the size and type of solar panel

•Voltage: Varies depending on the size and type of solar panel

•Current: Varies depending on the size and type of solar panel

- **Lithium Iron Battery** A lithium iron battery is a type of rechargeable battery that uses lithium iron phosphate (LiFePO4) as the cathode material. Lithium iron batteries are known for their long lifespan, high energy density, and safety.



International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 01 | January - 2024 SJIF Rating: 8.176 ISSN: 2582-3930

❖ **Specifications:**

•Nominal voltage: 3.2 V

•Nominal capacity: Varies depending on the size of the battery

•Cycle life: Varies depending on the size and manufacturer of the battery

SOFTWARE REQUIREMENTS

- **Arduino IDE:**

The Arduino IDE is a software application used to write and upload sketches to Arduino boards. It is a cross platform application, meaning that it can be used on Windows, macOS, and Linux computers. The Arduino IDE is made up of two main parts: the editor and the compiler. The editor is where you write your sketches. The compiler converts your sketches into machine code that can be executed by the Arduino board. The Arduino IDE also includes a number of other features, such as a serial monitor, a library manager, and a board manager. The serial monitor allows you to communicate with the Arduino board over a serial port. The library manager allows you to install and manage libraries. The board manager allows you to install and manage board definitions.

- **ESP32 Libraries:**

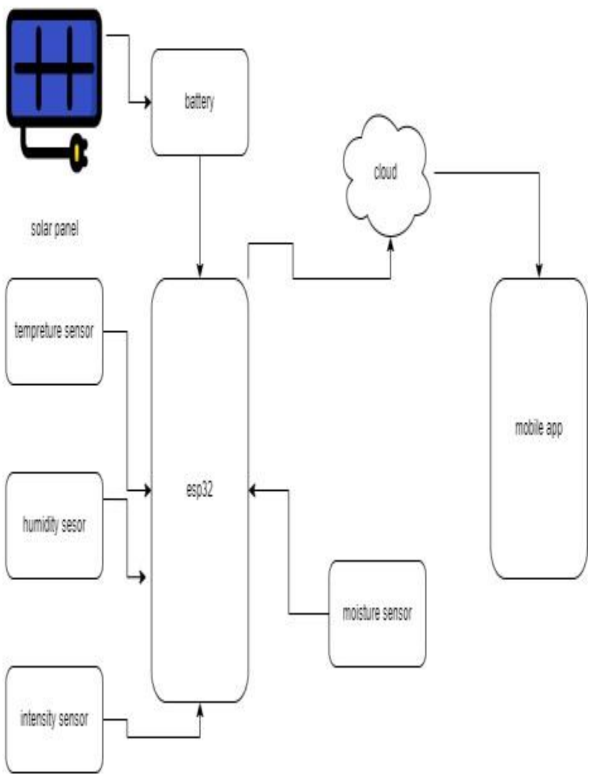
ESP32 libraries are software libraries that provide additional functionality to ESP32 boards. There are a wide variety of ESP32 libraries available, covering everything from Wi-Fi and Bluetooth to sensors and actuators. Some popular ESP32 libraries include:
•**WiFi.h:** Provides functions for connecting to and using Wi-Fi networks.
•**BluetoothSerial.h:** Provides functions for communicating with Bluetooth devices.
•**Wire.h:** Provides functions for communicating with I2C devices.
•**SPI.h:** Provides functions for communicating with SPI devices.
•**ADC.h:** Provides functions for reading analog values from the ESP32's ADC pins. •**DAC.h:** Provides functions for writing analog values to the ESP32's DAC pins.

- **Blynk Library:**

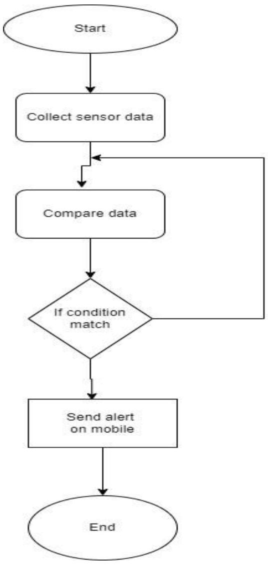
The Blynk library is an ESP32 library that allows you to connect your ESP32 board to the Blynk cloud. The Blynk cloud is a platform that allows you to create and control user interfaces for your IoT projects. To use the Blynk library, you will need to create an account on the Blynk website and create a new project. Once you have created a project, you will be given an auth token. You will need to add this auth token to your Arduino sketch and upload it to your ESP32 board. Once your ESP32 board is connected to the Blynk cloud, you can use the Blynk app to create and control user interfaces for your project. You can add widgets to your user interface to control your ESP32 board's GPIO pins, read sensor data, and send data to the cloud.

ARCHITECTURE DIAGRAM

- **ARCHITECTURE:**



- **FLOW CHART**



METHODOLOGIES

1. **Hardware setup:** •Connect the temperature sensor, humidity sensor, sun intensity sensor, and moisture sensor to the ESP32. •Connect a solar panel and battery to the ESP32 to power it. •Connect the ESP32 to the internet using a Wi-Fi module or Ethernet cable.

2. **Software setup:** •Install the Blynk app on your smartphone. •Create a Blynk project and add widgets for the temperature sensor, humidity sensor, sun intensity sensor, and moisture sensor. •Connect your Blynk project to the ESP32.
   •Train a machine learning model to predict plant diseases from sensor data. You can use a variety of machine learning algorithms, such as support vector machines, random forests, and convolutional neural networks. •Deploy the machine learning model to the ESP32.

3. **Operation:**
   •Place the ESP32 near the plant that you want to monitor. •The ESP32 will collect sensor data and send it to the Blynk app. •The Blynk app will display the sensor data in real time. •The machine learning model on the ESP32 will predict plant diseases from the sensor data. •If a plant disease is detected, the ESP32 will send an alert to the Blynk app.

CONCLUSION

This plant disease prediction system using IoT and ML can help farmers to detect plant diseases early and take corrective action to prevent crop losses. The system is relatively inexpensive and easy to implement, making it suitable for small-scale farmers.

REFERENCES

[1] Yi-Hua E. Yang, Hoang Le, Viktor K. Prasanna: High Performance Dictionary-Based String Matching for Deep Packet Inspection, IEEE 978-1-4244-5837-0.
[2] M. Roesch: Snort-Lightweight Intrusion Detection for Networks, 13th Systems Administration Conference, USENIX, 1999. [3] M. Roesch: Snort-Lightweight Intrusion Detection for Networks, 13th Systems Administration Conference, USENIX, 1999. [4] Yi-Hua E. Yang, Hoang Le, Viktor K. Prasanna: High Performance Dictionary-Based String Matching for Deep Packet Inspection, IEEE 978-1-4244-5837-0.