**IOT PROJECT REPORT**

****

**Name of the Student: University Roll No.:**

**Vaishnavi Sharma 12021002018038 (27)**

**Priyanka Kumari 12021002018041 (28)**

**Rajdeep Das 22022002018001 (54)**

Under the supervision of

Prof. Amartya Mukherjee

&

Prof. Tanima Bhowmik

***Academic Year: 2021-25***

**TITLE: AI-Driven Smart IoT Plant Monitoring System**

**REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING (COMPUTER SCIENCE AND BUSINESS SYSTEMS) OF**

**MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY**

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**(COMPUTER SCIENCE AND BUSINESS SYSTEMS)**

**INSTITUTE OF ENGINEERING AND MANAGEMENT**

**KOLKATA**

**ABSTRACT**

In modern agriculture, the integration of Internet of Things (IoT) technologies has revolutionized traditional farming practices, enabling more efficient and sustainable crop monitoring and management. This paper presents the design and implementation of an IoT-based Smart Agriculture System aimed at providing real-time monitoring of key environmental parameters crucial for crop health and yield optimization.

The system incorporates various sensors including air quality, soil moisture, and humidity sensors to continuously collect data pertaining to environmental conditions crucial for plant growth. These sensors are deployed strategically across the agricultural field to ensure comprehensive coverage and accurate data collection. The collected data is transmitted wirelessly to a central processing unit, where it is analyzed in real-time using advanced algorithms.

By leveraging IoT and data analytics, farmers gain valuable insights into the environmental conditions affecting their crops, allowing for timely interventions and optimized resource allocation. The system provides farmers with actionable information such as soil moisture levels, air quality metrics, and humidity levels, enabling them to make informed decisions regarding irrigation, fertilization, and pest control.

Furthermore, the system is equipped with remote monitoring and control capabilities, allowing farmers to access real-time data and control the agricultural parameters remotely through a user-friendly interface, accessible via smartphones or computers. This feature enhances operational efficiency and flexibility, empowering farmers to manage their crops effectively from anywhere, at any time.

Overall, the IoT-based Smart Agriculture System offers a comprehensive solution for modern agriculture, addressing the challenges of environmental variability and resource constraints. By providing real-time insights and remote monitoring capabilities, it enables farmers to optimize crop production, minimize resource wastage, and ultimately improve agricultural sustainability and productivity.

**INTRODUCTION**

The agricultural sector plays a pivotal role in ensuring food security and sustainable development worldwide. However, traditional farming practices are often challenged by environmental variability, resource constraints, and the increasing demand for higher crop yields. In response to these challenges, there is a growing interest in leveraging advanced technologies, such as the Internet of Things (IoT), to revolutionize agricultural practices and enhance productivity.

In this context, this paper introduces an IoT-based Smart Agriculture System designed to address the limitations of conventional farming methods by providing real-time monitoring and management of key environmental parameters critical for crop growth and development. The system integrates various sensors, including air quality, soil moisture, and humidity sensors, to collect data on environmental conditions within the agricultural field.

By continuously monitoring these parameters, farmers gain valuable insights into the health and status of their crops, enabling them to make informed decisions regarding irrigation, fertilization, and pest control. Moreover, the system offers remote monitoring and control capabilities, allowing farmers to access real-time data and manage agricultural parameters from anywhere, at any time.

Through the utilization of IoT technologies and data analytics, the Smart Agriculture System empowers farmers with the tools needed to optimize crop production, minimize resource wastage, and improve overall agricultural sustainability. By enhancing precision agriculture practices, this system contributes to the advancement of modern farming techniques, ultimately driving towards a more efficient, productive, and environmentally sustainable agricultural sector

**THEORY**

The IoT-based Smart Agriculture System relies on principles from several key areas to effectively monitor and manage crop environments.

1. **Sensor Technology**: Central to the system are various sensors deployed throughout the agricultural field. These sensors, including air quality, soil moisture, and humidity sensors, utilize principles of physics and chemistry to detect and measure environmental parameters critical for plant growth. For instance, soil moisture sensors employ capacitance or resistive techniques to measure water content in the soil, while air quality sensors may utilize gas-sensitive materials to detect pollutants or gases detrimental to plant health.
2. **Data Acquisition and Transmission**: Once sensor data is collected, it is transmitted wirelessly to a central processing unit. This data acquisition process involves principles of electronics and communication engineering, such as analog-to-digital conversion and wireless communication protocols (e.g., Wi-Fi, Zigbee, LoRa). Efficient data transmission ensures real-time monitoring capabilities, allowing farmers to receive timely updates on environmental conditions.
3. **Data Analytics and Decision Making**: Upon receiving sensor data, the central processing unit employs data analytics techniques to analyze and interpret the information. Machine learning algorithms, statistical models, and data visualization methods are applied to identify patterns, trends, and anomalies in the data. This process aids farmers in making informed decisions regarding irrigation scheduling, fertilizer application, and pest management strategies based on the analyzed data.
4. **Remote Monitoring and Control**: The system enables remote monitoring and control of agricultural parameters, allowing farmers to access real-time data and adjust environmental conditions from anywhere with internet connectivity. This functionality is facilitated by principles of computer science, networking, and user interface design. Through a user-friendly interface accessible via smartphones or computers, farmers can remotely monitor crop conditions and implement necessary interventions to optimize crop production.

By integrating principles from sensor technology, data acquisition and transmission, data analytics, and remote monitoring and control, the IoT-based Smart Agriculture System offers a comprehensive solution for modern farming practices. This interdisciplinary approach enables farmers to effectively monitor and manage crop environments, leading to improved agricultural productivity, resource efficiency, and sustainability.

**What is IOT?**

The Internet of Things (IoT) describes the network of physical objects—“things”—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet. These devices range from ordinary household objects to sophisticated industrial tools.

The field has evolved due to the convergence of multiple technologies, including ubiquitous computing, commodity sensors, increasingly powerful embedded systems, and machine learning.

Traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), independently and collectively enable the Internet of things. In the consumer market, IoT technology is most synonymous with products pertaining to the concept of the "smart home", including devices and appliances (such as lighting fixtures, thermostats, home security systems, cameras, and other home appliances) that support one or more common ecosystems, and can be controlled via devices associated with that ecosystem, such as smartphones and smart speakers. IoT is also used in healthcare systems.

There are a number of concerns about the risks in the growth of IoT technologies and products, especially in the areas of privacy and security, and consequently, industry and governmental moves to address these concerns have begun, including the development of international and local standards, guidelines, and regulatory frameworks.

IoT devices are a part of the larger concept of home automation, which can include lighting, heating and air conditioning, media and security systems and camera systems. Long-term benefits could include energy savings by automatically ensuring lights and electronics are turned off or by making the residents in the home aware of usage.

A smart toilet seat that measures blood pressure, weight, pulse and oxygen levels. A smart home or automated home could be based on a platform or hubs that control smart devices and appliances. For instance, using Apple's HomeKit, manufacturers can have their home products and accessories controlled by an application in iOS devices such as the iPhone and the Apple Watch. This could be a dedicated app or iOS native applications such as Siri. This can be demonstrated in the case of Lenovo's Smart Home Essentials, which is a line of smart home devices that are controlled through Apple's Home app or Siri without the need for a Wi-Fi bridge. There are also dedicated smart home hubs that are offered as standalone platforms to connect different smart home products and these include the Amazon Echo, Google Home, Apple's HomePod, and Samsung's SmartThings Hub. In addition to the commercial systems, there are many non-proprietary, open-source ecosystems; including Home Assistant, OpenHAB and Domoticz.

Significant numbers of energy-consuming devices (e.g. lamps, household appliances, motors, pumps, etc.) already integrate Internet connectivity, which can allow them to communicate with utilities not only to balance power generation but also helps optimize the energy consumption as a whole.These devices allow for remote control by users, or central management via a cloud-based interface, and enable functions like scheduling (e.g., remotely powering on or off heating systems, controlling ovens, changing lighting conditions, etc.).The smart grid is a utility-side IoT application; systems gather and act on energy and powerrelated information to improve the efficiency of the production and distribution of electricity.Usingadvanced metering infrastructure (AMI) Internet-connected devices, electric utilities not only collect data from end-users but also manage distribution automation devices like transformers.

Another example of integrating the IoT is Living Lab which integrates and combines research and innovation processes, establishing a public-private-people-partnership.There are currently 320 Living Labs that use the IoT to collaborate and share knowledge between stakeholders to co-create innovative and technological products. For companies to implement and develop IoT services for smart cities, they need to have incentives. The governments play key roles in smart city projects as changes in policies will help cities to implement the IoT which provides effectiveness, efficiency, and accuracy of the resources that are being used. For instance, the government provides tax incentives and cheap rent, improves public transport, and offers an environment where start-up companies, creative industries, and multinationals may co-create, share a common infrastructure and labor markets, and take advantage of locally embedded technologies, production process, and transaction costs.The relationship between the technology developers and governments who manage the city's assets is key to providing open access to resources to users in an efficient way.

In this project, we have tried to implement the concept of IoT to monitor the temperature, humidity and air quality and soil moisture for a plant

**COMPONENTS USED**

**HARDWARE USED: SOFTWARE USED:**

NodeMCU esp8266 – 2 ThingsPeak SaaS Cloud

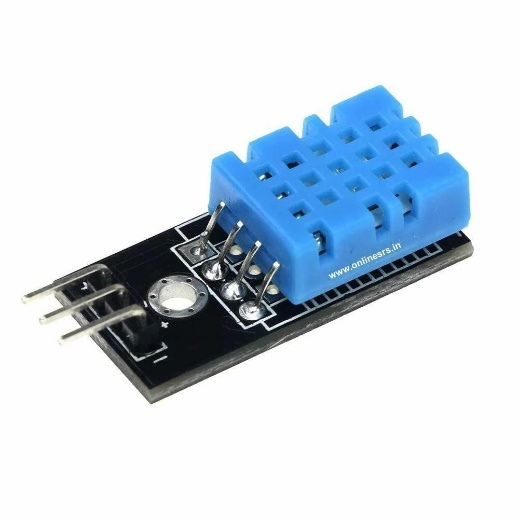
MQ135 sensor – 1 Arduino IDE

DHT11 sensor – 1 Google Colab

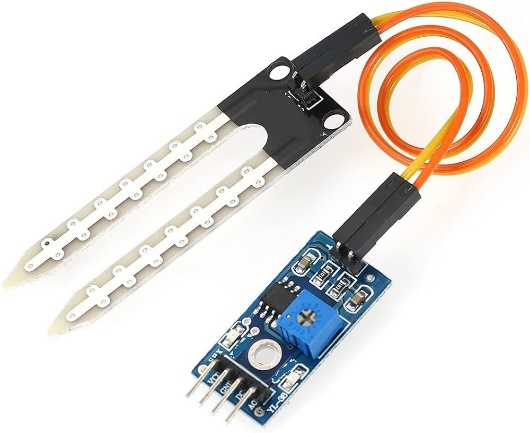
Soil Moisture sensor -1 HiveMQ

F2F Jumper Wires

NodeMCU Cables – 2

**DHT11 SENSOR:** The DHT11 sensor is widely utilized in IoT applications due to its simplicity, reliability, and affordability in measuring temperature and humidity. Its versatility finds application in various domains, including home automation, weather monitoring, smart agriculture, industrial monitoring, energy management, healthcare, and data logging. 

**MQ135 SENSOR:** The MQ135 gas sensor detects various harmful gases, including carbon dioxide, ammonia, benzene, and volatile organic compounds (VOCs), making it essential for air quality monitoring. Widely used in IoT applications, the MQ135 sensor aids in environmental monitoring, industrial safety, and indoor air quality assessment. Its versatility and sensitivity make it a valuable tool for detecting and mitigating the impact of air pollution in diverse settings. 

**Soil moisture sensors:** Soil moisture sensors are pivotal tools in agriculture and environmental monitoring, offering insights into soil conditions crucial for optimal plant growth and land management. These sensors measure the moisture content in the soil, providing valuable data for irrigation scheduling, water conservation, and crop health assessment. 

**NodeMCU ESP8266:** The NodeMCU ESP8266 is a versatile Wi-Fi development board based on the ESP8266 chipset, enabling IoT applications with ease of programming through the Arduino IDE or Lua scripting. Its compact size, integrated Wi-Fi connectivity, and GPIO pins make it ideal for projects requiring wireless communication and sensor integration, such as home automation, data logging, and remote monitoring. 

**SOFTWARE USED:**

**ThingSpeak:** ThingSpeak is an IoT platform that integrates seamlessly with the ESP8266 Wi-Fi module, enabling real-time data collection, visualization, and analysis. With ThingSpeak and the ESP8266, users can easily create IoT applications such as weather stations, home automation systems, and environmental monitoring solutions. The combination of ThingSpeak and the ESP8266 offers a convenient and powerful platform for building connected projects and gaining insights from sensor data.



**Arduino IDE:** The Arduino IDE is a user-friendly software tool for programming Arduino microcontrollers, offering a simple interface for writing, compiling, and uploading code. It includes built-in libraries and examples for beginners and advanced features like syntax highlighting and debugging tools for more experienced users. With its versatility, the Arduino IDE supports a wide range of electronic projects, from basic LED blinkers to advanced robotics applications. 

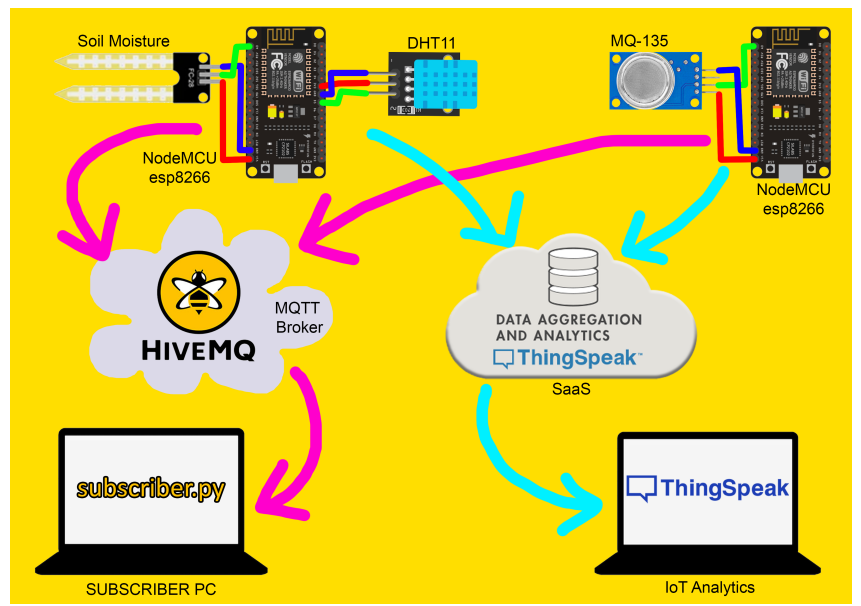
**HiveMQ:** HiveMQ is a scalable MQTT-based messaging platform designed for IoT applications, enabling efficient communication between devices and cloud infrastructure. With its robust features and high performance, HiveMQ facilitates real-time data exchange and management in large-scale IoT deployments. It offers seamless integration with existing systems and protocols, making it a preferred choice for enterprises and developers seeking reliable MQTT messaging solutions. 

**METHODOLOGY**

Our methodology incorporates several important elements which are crucial for the improvement of agricultural activities in space.

**IoT Environment Design and Sensor Integration**

The design phase of the AstroPlant Sentinel system starts with the creation of a customised IoT environment which is a pinnacle for space agriculture activities. Central to this endeavour is the integration of three essential sensors: soil moisture, DHT11 (to monitor temperature and humidity), and MQ135 (for reflection of air quality). That is, these sensors are calibrated for certain environmental conditions of space and form the foundation of our surveillance system

****

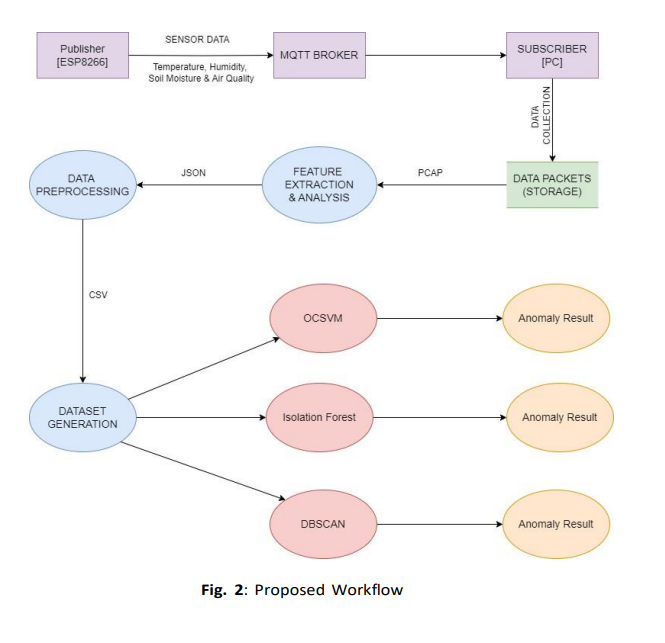
**Data Preprocessing**

The Obtained data by the sensors, are sent from the microcontroller of the NodeMCU to HiveMQ broker from which to the subscriber PC is relayed[11]. That is where a subscriber.py script will grab and decode using Wireshark that saves the packets in the format of“.pcap”. Consequently, these masqueraded packets go through a preprocessing step, where process features are selected and extracted [8]. In particular, those specified features such as “frame.time” and “mqtt.msg” which are drawn out from the data analysis. This preprocessing procedure is of critical importance, in order to accomplish the purification of data during the analysis step.

**Anomaly Detection System (ADS)**

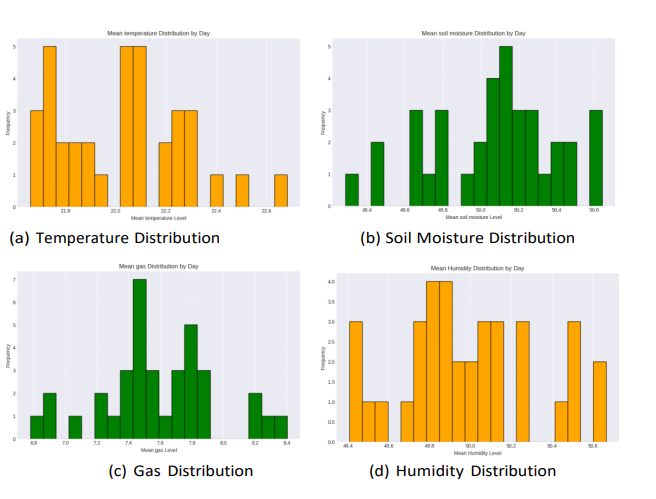
Development Recognizing the critical importance of anomaly detection in safeguarding plant health and productivity in space, our research focuses on the development and optimization of a dedicated ADS. Leveraging machine learning techniques, specifically One-Class Support Vector Machines (OCSVM), Isolation Forest, and Density-Based Spatial Clustering of Applications with Noise (DBSCAN), we aim to identify and mitigate anomalies in environmental data streams generated by the IoT sensors.

**Data Visualization and Integration with Cloud Platforms**

Data Visualization in space agriculture monitoring is of critical importance in enabling the interpretation and the decision-making process. To this end, our tasks encompass building intuitive visualisation tools using SaaS cloud-based platforms. We intend to achieve this by means of combining sensor data streams connected on SaaS platforms like ThingSpeak that will allow mission controllers to access up-to-date information about the space shuttle or extraterrestrial habitats. **Anomaly Detection System (ADS)**

Development Recognizing the critical importance of anomaly detection in safeguarding plant health and productivity in space, our research focuses on the development and optimization of a dedicated ADS. Leveraging machine learning techniques, specifically One-Class Support Vector Machines (OCSVM), Isolation Forest, and Density-Based Spatial Clustering of Applications with Noise (DBSCAN), we aim to identify and mitigate anomalies in environmental data streams generated by the IoT sensors.

**Data Visualization and Integration with Cloud Platforms**

Data Visualization in space agriculture monitoring is of critical importance in enabling the interpretation and the decision-making process. To this end, our tasks encompass building intuitive visualisation tools using SaaS cloud-based platforms. We intend to achieve this by means of combining sensor data streams connected on SaaS platforms like ThingSpeak that will allow mission controllers to access up-to-date information about the space shuttle or extraterrestrial habitats

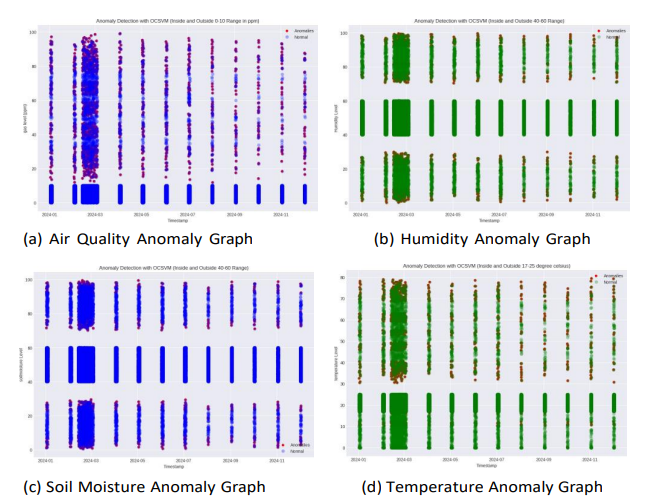
**Real-Time Anomaly Detection Implementation**

Broadly, to expand our initiative of real-time anomaly detection, we have devised a plan that complements our annotated dataset. Primarily, a programme has been written which connects the broker with our sensors to subscribe to the sensor data is the initial step for the script development. Script features fast data capturing and live storing capabilities; incoming data will be saved in .csv dataset file as well as time of capturing of data will be logged down too. Therefore, this dataset will necessarily be extended in real time like this script will work constantly. Along with the data collection, another script will be designed for accessing the above trained machine learning model. This script will keep track of records that are in the dataset daringly waiting for new data. When data arrives, the model script ML algorithm will right away come into play by detecting any anomalies in real-time, prompting quick and timely responses to evolving environmental conditions. Having a system integrated with these functions, we envisage a complete autonomous anomaly detection system that operates in real time and therefore increases the level of efficiency and reliability in space agriculture monitoring aboard space missions in the future.

**Results Evaluation**

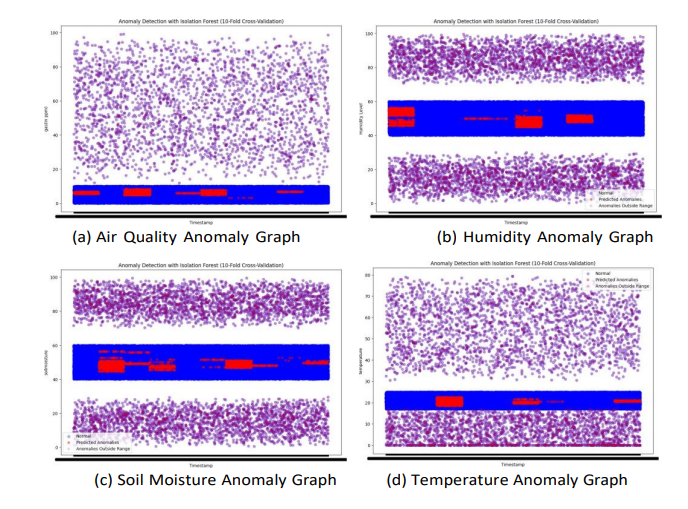
In our research on anomaly detection forspace agriculture monitoring using the Astro- Plant Sentinel system, we conducted extensive evaluations of three machine learning models: One-Class Support Vector Machine (OCSVM), Isolation Forest, and Density- Based Spatial Clustering of Applications with Noise (DBSCAN). Each model was assessed based on its performance in detecting anomalies across four types of sensor data: humidity, temperature, soil moisture, and gas concentration.

**One-Class Support Vector Machine (OCSVM)**

OCSVM is a very convenient algorithm to use when there is a need to detect unknown instances in a dataset in which most of the data is classified as ‘normal’. It is based upon the use of a boundary that results in crowding and finds deviations across this boundary as misconduct. Accuracy and recall rate of OCSVM remain the most consequential factors for its performance, which can be greatly influenced by the hyperparameters selection and abnormal instances distribution. 

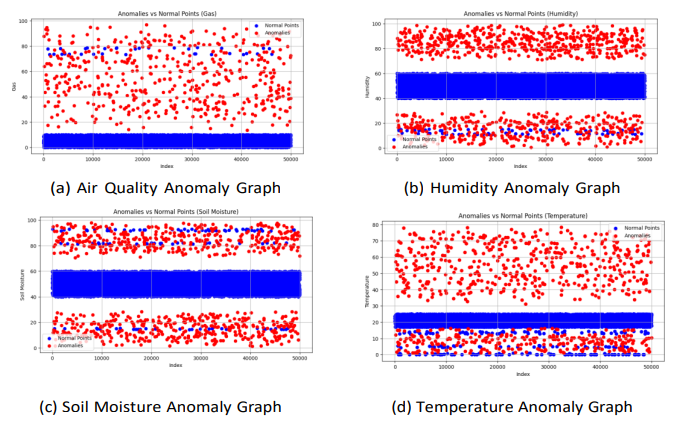
**Isolation Forest**

The Isolation Forest has high efficiency in the detection of anomalies in high- dimensional datasets. It stands out and separates instances from regions of feature space where there is a low density of instances. It does so by recursively slicing the data into smaller, increasingly homogeneous sets in which anomalies are usually con- tained in a smaller number of sets. Importantly, Isolation Forest not only shows high accuracy but its performance on recall rate is relatively low because Isolation Forest concentrates its power on accuracy of anomalies instead of the full distinction between the normal and the abnormal cases.



**Density-Based Spatial Clustering of Applications with Noise**

The DBSCAN is a density-based clustering algorithm that is adept at detecting clus- ters with different shapes and sizes as well as deficient point noise. It operates by running the nearby points in clusters as well as indicating noise, which makes it an appropriate algorithm to employ in the detection of anomalies. The great performance of DBSCAN in automatically tracking and reporting of anomalies is due to its abil- ity to fit the general data density pattern, through which it detects anomalies across different sensor types



**IMPLEMENTATION**

**code:**

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

#include <DHT.h>

String apiKey = "TQV7IORN1TPY61A7"; // Enter your Write API key from ThingSpeak

String channelId = "2470641"; // Enter your ThingSpeak channel ID

const char\* ssid = "rajdeep";

const char\* password = "kenn9870"; // Replace with your WiFi password

const char \*thingspeakServer = "api.thingspeak.com";

const char \*mqttServer = "broker.hivemq.com";

const char \*mqttTopic1 = "temp\_astroplant";

const char \*mqttTopic2 = "hu\_astroplant";

const char \*mqttTopic3 = "air\_astroplant";

#define DHTPIN D5 // Pin where the DHT11 is connected D1

#define DHTTYPE DHT11 // Type of DHT sensor

#define MQ135\_PIN A0 // Pin where the MQ135 sensor is connected

#define LED\_PIN D2 // Pin where the LED is connected

WiFiClient wifiClient;

PubSubClient mqttClient(wifiClient);

DHT dht(DHTPIN, DHTTYPE);

unsigned long lastMQTTPublishTime = 0;

unsigned long lastThingSpeakUpdateTime = 0;

void setup() {

Serial.begin(115200);

pinMode(LED\_PIN, OUTPUT); // Set the LED pin as an output

digitalWrite(LED\_PIN, LOW); // Turn off the LED initially

delay(10);

Serial.println("Connecting to WiFi...");

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

mqttClient.setServer(mqttServer, 1883);

}

void loop() {

if (!mqttClient.connected()) {

reconnectMQTT();

}

mqttClient.loop();

unsigned long currentMillis = millis();

// Publish to MQTT every 3 seconds

if (currentMillis - lastMQTTPublishTime >= 3000) {

publishMQTT();

lastMQTTPublishTime = currentMillis;

}

// Update ThingSpeak every 7 seconds

if (currentMillis - lastThingSpeakUpdateTime >= 7000) {

updateThingSpeak();

lastThingSpeakUpdateTime = currentMillis;

}

}

void publishMQTT() {

float temperature = dht.readTemperature(); // Read temperature from DHT11

float humidity = dht.readHumidity(); // Read humidity from DHT11

int gasValue = analogRead(MQ135\_PIN);

float gasConcentration = map(gasValue, 0, 1023, 0, 1000);

String mqttPayload1 = String(temperature, 2);

String mqttPayload2 = String(humidity, 2);

String mqttPayload3 = String(gasConcentration, 2);

mqttClient.publish(mqttTopic1, mqttPayload1.c\_str());

mqttClient.publish(mqttTopic2, mqttPayload2.c\_str());

mqttClient.publish(mqttTopic3, mqttPayload3.c\_str());

Serial.print("Temperature: ");

Serial.print(temperature);

Serial.println(". Sent to MQTT.");

Serial.print("Humidity: ");

Serial.print(humidity);

Serial.println(". Sent to MQTT.");

Serial.print("Air Quality: ");

Serial.print(gasConcentration);

Serial.println(". Sent to MQTT.");

// Turn on the LED for a second

digitalWrite(LED\_PIN, HIGH);

delay(1000);

digitalWrite(LED\_PIN, LOW);

}

void updateThingSpeak() {

float temperature = dht.readTemperature(); // Read temperature from DHT11

float humidity = dht.readHumidity(); // Read humidity from DHT11

int gasValue = analogRead(MQ135\_PIN);

float gasConcentration = map(gasValue, 0, 1023, 0, 1000);

if (WiFi.status() == WL\_CONNECTED) {

if (wifiClient.connect(thingspeakServer, 80)) {

String postStr = apiKey;

postStr += "&field1=";

postStr += String(temperature, 2); // Add temperature to field 1

postStr += "&field2=";

postStr += String(humidity, 2); // Add humidity to field 2

postStr += "&field4=";

postStr += String(gasConcentration, 2); // Add air quality to field 3

postStr += "\r\n\r\n";

wifiClient.print("POST /update HTTP/1.1\n");

wifiClient.print("Host: api.thingspeak.com\n");

wifiClient.print("Connection: close\n");

wifiClient.print("X-THINGSPEAKAPIKEY: " + apiKey + "\n");

wifiClient.print("Content-Type: application/x-www-form-urlencoded\n");

wifiClient.print("Content-Length: ");

wifiClient.print(postStr.length());

wifiClient.print("\n\n");

wifiClient.print(postStr);

Serial.print("Temperature: ");

Serial.print(temperature);

Serial.println(". Sent to Thingspeak.");

Serial.print("Humidity: ");

Serial.print(humidity);

Serial.println(". Sent to Thingspeak.");

Serial.print("Air Quality: ");

Serial.print(gasConcentration);

Serial.println(". Sent to Thingspeak.");

// Turn on the LED for a second

digitalWrite(LED\_PIN, HIGH);

delay(1000);

digitalWrite(LED\_PIN, LOW);

}

wifiClient.stop();

}

}

void reconnectMQTT() {

while (!mqttClient.connected()) {

Serial.print("Attempting MQTT connection...");

if (mqttClient.connect("ESP8266Client")) {

Serial.println("connected");

} else {

Serial.print("failed, rc=");

Serial.print(mqttClient.state());

Serial.println(" try again in 5 seconds");

delay(5000);

}

}

}

**Conclusion**

The IoT-based Smart Agriculture System represents a significant advancement in modern farming practices, offering a comprehensive solution for monitoring and managing crop environments in real-time. Through the integration of sensor technology, data analytics, and remote monitoring capabilities, the system empowers farmers with the tools needed to optimize crop production, minimize resource wastage, and enhance agricultural sustainability.

By continuously monitoring key environmental parameters such as air quality, soil moisture, and humidity, the system provides farmers with actionable insights into the health and status of their crops. This information enables informed decision-making regarding irrigation scheduling, fertilizer application, and pest control strategies, ultimately leading to improved crop yields and quality.

Furthermore, the remote monitoring and control features of the system offer farmers unprecedented flexibility and accessibility in managing their agricultural operations. With the ability to access real-time data and adjust environmental conditions remotely via a user-friendly interface, farmers can effectively respond to changing crop needs and environmental conditions from anywhere, at any time.

Overall, the IoT-based Smart Agriculture System not only enhances operational efficiency and productivity but also contributes to the promotion of sustainable farming practices. By optimizing resource utilization and minimizing environmental impact, the system supports the long-term viability of agriculture while ensuring food security and economic prosperity for farming communities worldwide. As technology continues to evolve, the integration of IoT solutions in agriculture holds immense promise for addressing the challenges of feeding a growing global population in a sustainable and efficient manner.

Top of Form