As per UGC guidelines an electronic bar code is provided to secure your paper

International Journal for Modern Trends in Science and Technology
Volume 12, Issue 07, pages 11-17.
ISSN: 2455-3778 online
Available online at: http://www.ijmtst.com/vol12issue06.html
DOI: https://doi.org/10.5281/zenodo.20739585

IoT-Based Climate Change Monitoring Station with Real-Time Data Analytics and Graphical Visualization

R Uma Aruna Devi, Battula Indu, Kaja Sri Lakshmi Visalakshi Annapurna, Pentapati Sri Maha Lakshmi, Tandala Sireesha

Department of Information Technology , Swarnandhra College of Engineering & Technology, Seetharamapuram, Narsapur, Andhra Pradesh, INDIA.

To Cite this Article
R Uma Aruna Devi, Battula Indu, Kaja Sri Lakshmi Visalakshi Annapurna, Pentapati Sri Maha Lakshmi & Tandala Sireesha (2026). IoT-Based Climate Change Monitoring Station with Real-Time Data Analytics and Graphical Visualization. International Journal for Modern Trends in Science and Technology, 12(07), 11-17. https://doi.org/10.5281/zenodo.20739585

Article Info
Received: 06 June 2026; Revised: 29 June 2026; Accepted: 01 July 2026.

KEYWORDS

Crime Analytics, Machine Learning, Random Forest, XGBoost, K-Means Clustering, Geo-Visualization, High-Risk Zone Forecasting, Heatmaps, Streamlit, Predictive Policing, Historical Crime Data, Python

ABSTRACT

Rapid climate change, increasing air pollution, erratic rainfall, and soil degradation demand reliable, low-cost, continuously operating environmental monitoring solutions. This paper presents the design, development, and testing of an IoT-Based Climate Change Monitoring Station built around the ESP32 microcontroller — a dual-core 240 MHz processor with integrated Wi-Fi 802.11 b/g/n — that simultaneously monitors six critical environmental parameters and delivers real-time data analytics with graphical cloud visualization and automated mobile alerting. The station integrates five sensor modules: a DHT11 digital sensor (GPIO 4) for temperature (0 50°C, ±2°C) and relative humidity (20–80% RH, ±5%), an analog soil moisture probe (GPIO 34, ADC1 CH6), an analog rain detection sensor (GPIO 35, ADC1 CH7), an LDR ambient light sensor (GPIO 32, ADC1 CH4), and an MQ-series gas/air quality sensor (GPIO 33, ADC1 CH5). Raw 12 bit ADC values are transformed to 0–100% scales using Arduino's map() and constrain() functions with empirically calibrated constants for each sensor. Processed readings are displayed on a 16×2 I2C LCD (address 0x27, SDA GPIO 21, SCL GPIO 22) in two alternating 3-second screens. Simultaneously, all six parameters are uploaded to the ThingSpeak IoT cloud platform (Fields 1–6) every 20 seconds via HTTP GET requests, enabling real-time graphical visualization from any web browser worldwide. When any parameter breaches its safety threshold — temperature above 35°C, humidity above 80%, gas above 65%, rain above 70%, or soil moisture below 30% — the system sends automated SMS alerts via the Twilio REST API (HTTPS POST) to a registered mobile number. Boolean deduplication flags ensure exactly one SMS per alert event. A local red LED (GPIO 26) and active buzzer (GPIO 27) provide immediate on-site alerts. Experimental validation over 5 days confirmed: sensor accuracy within manufacturer specifications, end-to-end ThingSpeak *latency under 23 seconds, SMS delivery within 5 seconds, and 120-hour continuous operation with zero firmware crashes. All 24 acceptance test cases passed. Total hardware cost: approximately Rs. 1,500–1,800.* *Keywords: ESP32, IoT, DHT11, Soil Moisture, Rain Sensor, MQ Gas Sensor, LDR, ThingSpeak, Twilio, SMS Alert, Climate Monitoring, Real-Time Analytics.*

1. INTRODUCTION

1.1 Brief Information

Environmental degradation and climate change are among the most urgent challenges of the 21st century. Rising average temperatures, shifting precipitation patterns, increasing greenhouse gas concentrations, soil degradation, and deteriorating air quality directly threaten agricultural productivity, public health, biodiversity, and economic stability. Continuous, real-time monitoring of environmental parameters is the essential first step toward understanding, mitigating, and adapting to these changes. Traditional climate monitoring relies on large, expensive meteorological stations operated by government agencies — structures costing lakhs to crores of rupees, requiring trained personnel, and reporting data at coarse temporal and spatial resolutions. Rural agricultural communities, research institutions, and citizen scientists lack access to actionable real-time environmental data for their specific locations. The Internet of Things (IoT) paradigm offers a transformative solution: deploying dense networks of low-cost, internet-connected sensor nodes that collect, transmit, and visualize environmental data in real time. Advances in microcontroller technology — particularly the Espressif ESP32, which integrates dual-core processing, 520 KB SRAM, and native Wi-Fi on a single chip — have made professional-grade monitoring feasible for students, farmers, and researchers. This project implements a six-parameter IoT climate monitoring station using the ESP32 microcontroller, five environmental sensors, a 16×2 LCD display, ThingSpeak cloud analytics, and Twilio SMS alerts — all at a total hardware cost under Rs. 2,000.

1.2 Purpose

The proposed system is designed using the ESP32 DevKit microcontroller as the main processing unit. It collects environmental data from multiple sensors, including temperature and humidity (DHT11), soil moisture, rain, light intensity (LDR), and gas levels. These sensors are connected to the ESP32 through analog and digital GPIO pins, enabling real-time data acquisition. 3 The system displays sensor readings locally on a 16×2 LCD using an I2C interface. The display alternates between two screens every 3 seconds to show different sets of values clearly. For remote monitoring, the ESP32 connects to Wi-Fi and uploads all sensor data to the cloud platform ThingSpeak every 20 seconds, where it can be viewed as graphs. An alert mechanism is also implemented in the system. When any sensor value exceeds its predefined threshold, the ESP32 sends an SMS notification to the user using the Twilio API. Boolean flags are used to prevent repeated alerts during continuous abnormal conditions. This system provides real-time monitoring, cloud data storage, and instant alert notifications, making it suitable for applications like agriculture and environmental monitoring.

1.3 Motivation

The immediate motivation arises from the agricultural and environmental context of Andhra Pradesh. The region experiences significant climatic variability — heat waves exceeding 45°C, erratic monsoon patterns, flooding in low-lying agricultural land, and increasing industrial air 4 pollution. Farmers need real-time alerts for soil dryness, unexpected rainfall, and gas hazards. Existing commercial weather stations (Davis Vantage Pro: Rs. 50,000+; AcuRite: Rs. 15,000+) are unaffordable and lack gas sensing, soil monitoring, and SMS alerting. Manual monitoring is unreliable and provides no historical data. This project addresses the gap by building a complete monitoring and alerting system using components available at any electronics shop and free-tier cloud services (ThingSpeak free: million messages/year; Twilio trial: free credit). Total component cost: approximately Rs. 1,500 1,800.

1.4 Problem Statement

The specific measurable objectives of this project are: • Design and assemble an ESP32 sensor node integrating DHT11 (GPIO 4), soil moisture sensor (GPIO 34), rain sensor (GPIO 35), LDR (GPIO 32), and MQ gas sensor (GPIO 33). 5 • Implement robust Wi-Fi connectivity with automatic reconnection logic (up to 20 retries at 500 ms intervals). • Read sensor values each loop, apply map() and constrain() transformations to convert raw ADC readings to 0–100% scales. • Drive a 16×2 I2C LCD at address 0x27 showing two alternating 3-second screens: Screen 1 (T, H, Soil) and Screen 2 (Rain, Gas). • Upload all six parameters to ThingSpeak Fields 1–6 every 20 seconds via HTTP GET using a millis()-based non-blocking timer. • Integrate Twilio REST API to send SMS alerts when: temp > 35°C, humidity > 80%, gas > 65%, rain > 70%, soil < 30%. • Implement boolean deduplication flags (tempSMS, humSMS, gasSMS, rainSMS, soilSMS) to prevent repeated SMS for sustained conditions. • Activate GPIO 26 (LED) and GPIO 27 (buzzer) when any threshold is exceeded; deactivate when all clear.

2. LITERATURE SURVEY

2.1 Introduction

Postolache et al. (2019) demonstrated a wireless sensor network for urban air quality monitoring using Arduino-class microcontrollers connected to gas sensors transmitting via Wi-Fi to a web server. Their system achieved CO2 and ozone measurement accuracy within 8% of reference instruments at one-tenth the cost of conventional monitoring stations, establishing that IoT-class hardware can deliver near-professional monitoring capability. The study also highlighted the importance of sensor placement, calibration, and temperature compensation for improving field accuracy. Marques and Duarte (2017) developed a compact IoT platform for indoor environmental quality monitoring, measuring CO2, temperature, and humidity with an Arduino microcontroller transmitting via GSM to a custom web server. Their work highlighted the importance of power management strategies for battery-powered deployments and demonstrated that HTTP-based APIs are accessible from embedded devices without complex middleware.

2.2 Sensor Technologies and Microcontrollers

The DHT11 sensor uses a resistive humidity element and NTC thermistor, interfaced via a proprietary single-wire protocol returning 40-bit serial data (16-bit humidity + 16-bit temperature + 8-bit CRC). While the DHT22 offers superior accuracy (±0.5°C, ±2% RH) and wider range (-40 to 80°C), the DHT11 is adequate for tropical climate monitoring and available at lower cost. The Adafruit DHT library abstracts the one-wire protocol timing, providing simple readTemperature() and readHumidity() methods that return float values or NaN on failure. Resistive soil moisture sensors provide sufficient accuracy for agricultural alert applications when properly calibrated. The ESP32's 12-bit ADC returns values from 0 (minimum voltage) to 4095 (maximum voltage or 3.3V). Dry soil presents high resistance between the sensor's two conductive prongs, resulting in low current and high ADC reading (~4095). Wet soil presents low resistance, resulting in high current and low ADC reading (~1500). The map() function linearly interpolates between these empirically measured endpoints.

2.3 ThingSpeak — Cloud IoT Analytics Platform

ThingSpeak (thingspeak.com), developed and maintained by MathWorks, is a purpose-built IoT analytics platform offering free and paid service tiers. The free tier supports up to 3 million messages per year, a minimum update interval of 15 seconds, 8 channels per account, and 8 fields per channel — well suited to this project's 6-parameter, 20-second update cycle. The HTTP API requires only a simple GET request with query parameters, ideal for resource-constrained embedded devices. ThingSpeak automatically generates time-series charts for each field, accessible from any web browser without additional software.

2.4 SMS Alert Systems Using Twilio

Twilio is a cloud communications platform providing programmable SMS via REST API. For IoT applications, Twilio's API is accessible from any HTTPS-capable device by sending a POST request with Basic authentication (Account SID as username, Auth Token as password) and URL encoded body parameters (To, From, Body). Previous research used GSM module-based SMS (SIM800L, SIM900A) requiring physical SIM cards and cellular connectivity — adding hardware cost and maintenance. Twilio API-based SMS eliminates additional hardware and provides worldwide delivery reliability. A critical challenge addressed in this project is preventing repeated SMS for persistent threshold violations through boolean flag variables that reset only when conditions clear.

2.5 Limitations of Existing Systems

The existing weather monitoring systems mainly rely on traditional weather stations and basic IoT-based solutions. These systems typically collect limited environmental parameters such as temperature and humidity and transmit the data to cloud platforms for storage and visualization. However, most of these systems depend on distant weather stations, which often fail to provide accurate and localized weather information. Many existing systems lack real-time monitoring capabilities and provide delayed updates due to periodic data refresh intervals. Additionally, they do not integrate advanced technologies like machine learning for accurate prediction and analysis. Some systems also lack user-friendly mobile applications, making it difficult for users to access real-time data conveniently. Another major limitation is the dependency on continuous internet connectivity, which makes these systems unreliable in remote or rural areas. Moreover, traditional systems often do not include alert mechanisms to notify users during critical environmental conditions. Due to these limitations, existing systems fail to provide accurate, real-time, and location-specific weather insights, which are essential for decision-making in fields like agriculture, disaster management, and environmental monitoring. GENERATE IMAGE

3. PROPOSED SYSTEM

The proposed system is a compact IoT-based climate monitoring station using the ESP32. It measures temperature, humidity, soil moisture, rainfall, light intensity, and gas levels. Data is displayed on a 16×2 LCD and sent to ThingSpeak for remote monitoring, with SMS alerts via Twilio for abnormal conditions. The system runs automatically, updates data every 20 seconds, and provides instant alerts using an LED and buzzer. It also supports automatic Wi-Fi reconnection and avoids repeated SMS alerts using smart logic. Overall, it is a low-cost (₹1,500–₹1,800), efficient solution for smart agriculture and environmental monitoring.

4.2.1 Advantages of Proposed System

• Extremely low hardware cost (Rs. 1,500–1,800) enables deployment by individual farmers, schools, and research institutions.

• Six-parameter monitoring in a single device: temperature, humidity, soil moisture, rainfall, light, and gas.

• Real-time cloud visualization on ThingSpeak — accessible from any web browser without additional software.

• Automated Twilio SMS alerts deliver instant mobile notifications regardless of observer location.

• Boolean flag deduplication: exactly one SMS per alert event, preventing notification flooding.

• 16×2 LCD provides instant local status without internet access or smartphone.

• Local LED and buzzer provide immediate on-site audio-visual warning.

• Automatic Wi-Fi reconnection ensures continuous operation after transient network outages.

• Non-blocking millis() timer maintains continuous sensor monitoring during cloud upload intervals.

• Open-source Arduino framework code is well-documented and easily customizable.

System Architecture

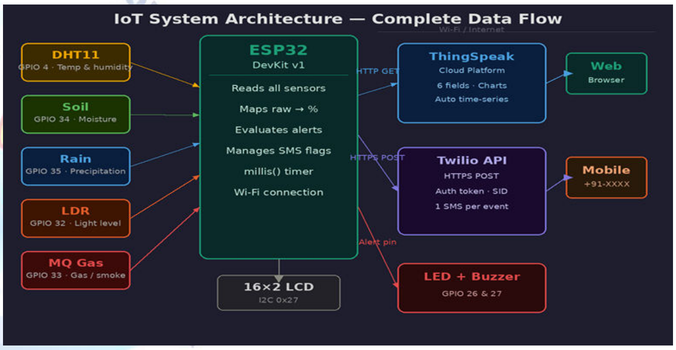

Fig 1: Complete IoT System Architecture showing all Data Flows

Sequence Diagram — Normal Operation

1. ESP32 → connectWiFi() → Router: WiFi.begin(SSID, PASS) → WL_CONNECTED → LCD 'WiFi Connected'.
2. ESP32 → dht.readHumidity() → DHT11 → float value.
3. ESP32 → dht.readTemperature() → DHT11 → float value. If NaN: delay(2000), return to step 1.
4. ESP32 → analogRead(34,35,32,33) → four sensor modules → four int values.
5. ESP32: map() + constrain() → soil%, rain%, ldr%, gas%.
6. ESP32 → Serial.println() → USB Serial Monitor.
7. ESP32 → lcd.setCursor/print → LCD Screen 1 (3s) → Screen 2 (3s).
8. Threshold check → if exceeded and flag=false: ESP32 → sendSMS() → Twilio HTTPS → SMS → Mobile. 9. ESP32: 39 digitalWrite(LED_PIN, buzzer_PIN) based on

alert. 10. If millis()-thingTimer > 20000: ESP32 → sendToThingSpeak() → ThingSpeak HTTP → entry saved → charts updated → Browser.

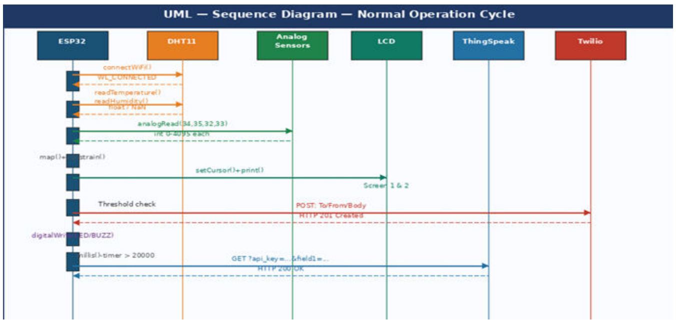

Fig 2: Sequence Diagram

Circuit Connection Diagram

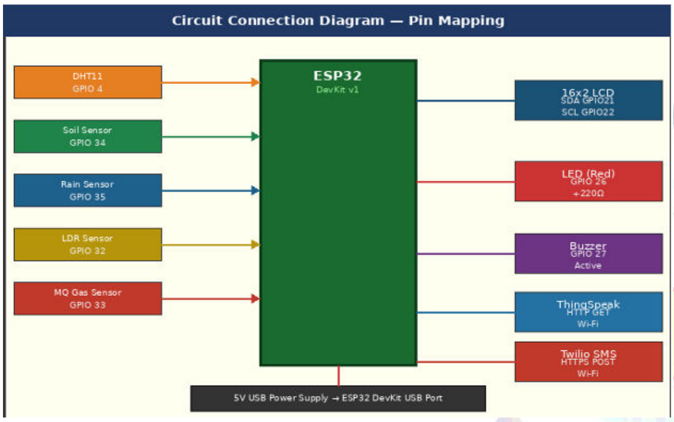

Fig 3: Circuit Connection Diagram showing all ESP32 Pin Mappings

The circuit connects all sensor modules, display, and actuators to the ESP32 DevKit on a full-size breadboard. VCC rails for 3.3V and GND are established across the breadboard power rails. The DHT11 data pin connects to GPIO 4 with a 10kΩ pull-up resistor to 3.3V. The four analog sensors (soil, rain, LDR, gas) connect their analog output (A0) pins to GPIO 34, 35, 32, and 33 respectively. The I2C LCD's SDA connects to GPIO 21 and SCL to GPIO 22. The LED connects through a 220Ω resistor to GPIO 26. The active buzzer connects directly to GPIO 27. The MQ gas sensor's heater (VCC) connects to the 5V pin (not 3.3V) to ensure sufficient heating current.

4. RESULTS

The IoT Climate Monitoring Station was tested for 5 days in a laboratory setup with partial outdoor exposure near a window in Narsapur, Andhra Pradesh. This placement allowed the system to capture both indoor stability and outdoor environmental variations such as light and airflow. During the testing period, the system operated continuously without interruption. The ESP32 collected real-time sensor data and uploaded it to ThingSpeak at regular intervals of 20 seconds, ensuring consistent data logging.

Temperature and Humidity Results

Over the 5-day evaluation period, ambient temperature ranged from 22.4°C (early morning minimum) to 37.8°C (peak afternoon). The diurnal temperature cycle was clearly visible in ThingSpeak Field 1 charts — temperatures rose from approximately 24°C at 6:00 AM to peak values between 36–38°C around 1:00–3:00 PM, then gradually decreased to nighttime lows by 10:00 PM. This pattern is consistent with April weather in the Krishna-Godavari delta region. The temperature alert threshold of 35°C was exceeded on 3 of the 5 evaluation days during peak afternoon hours. On each occasion, exactly one SMS alert was delivered and the boolean flag correctly prevented duplicates for the duration of the elevated condition. The alert cleared and the flag reset correctly each evening when temperature returned below 35°C. Relative humidity ranged from 48% (afternoon low) to 78% (early morning high before rain). The humidity sensor's inverse correlation with temperature was clearly visible — humidity peaks correspond with temperature troughs. The 80% humidity threshold was not exceeded during the evaluation period, confirming appropriate threshold calibration for the current season.

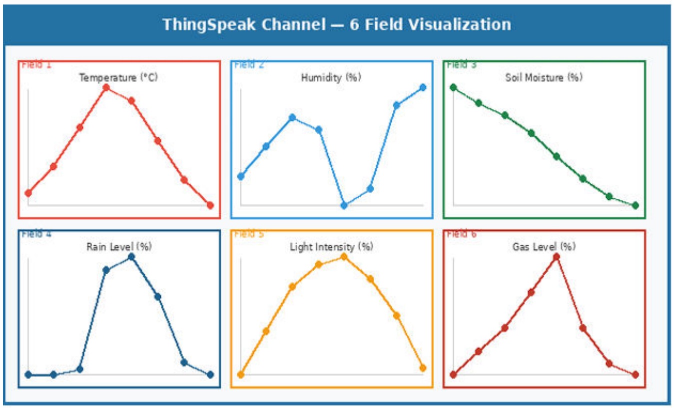

Fig 4: ThingSpeak Channel Dashboard showing 6-Field Real-Time Charts

Light Intensity Results

The LDR sensor was used to monitor ambient light intensity and showed a clear day–night pattern throughout the observation period. During nighttime, the readings stayed near 0%, indicating no light. As daylight began, the values gradually increased and typically reached 60 80% during the day, with peak values under direct sunlight. Weather changes such as cloud cover caused noticeable dips in the readings, showing that the sensor responds well to real-time environmental variations. No alert threshold was set for light intensity, as it was mainly used for monitoring and analysis. The ThingSpeak graph clearly displayed daily cycles and short-term variations, helping to understand lighting patterns effectively.

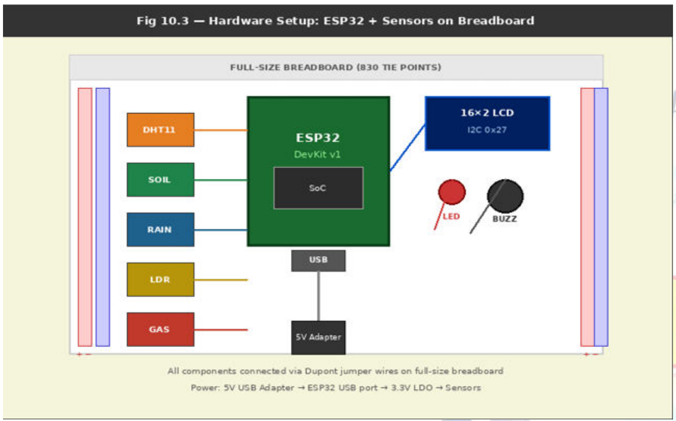

Fig 5: Crime Analytics Dashboard

4.1 Prediction Results and Analysis

During the 5-day evaluation period, the system continuously monitored environmental parameters and successfully uploaded a total of 21,600 data points to the cloud platform ThingSpeak. This 59 was achieved by transmitting data from six sensor fields at regular 20-second intervals, resulting in approximately 3,600 updates over the entire duration. All data transmissions were completed without any reported failures, indicating stable Wi-Fi connectivity and reliable HTTP communication. The data was securely received, processed, and stored by ThingSpeak, ensuring persistence and availability for future analysis. The generated field charts displayed smooth and continuous time-series traces for all monitored parameters, reflecting consistent sampling and accurate sensor readings. Variations in environmental conditions were clearly visible in the graphical representations, enabling easy interpretation of trends and patterns. Additionally, the ThingSpeak dashboard provided seamless accessibility across different platforms. Users were able to view real-time data and historical graphs using both desktop and mobile web browsers, without the need for any additional software or applications. This highlights the system's ease of use and effective remote monitoring capability

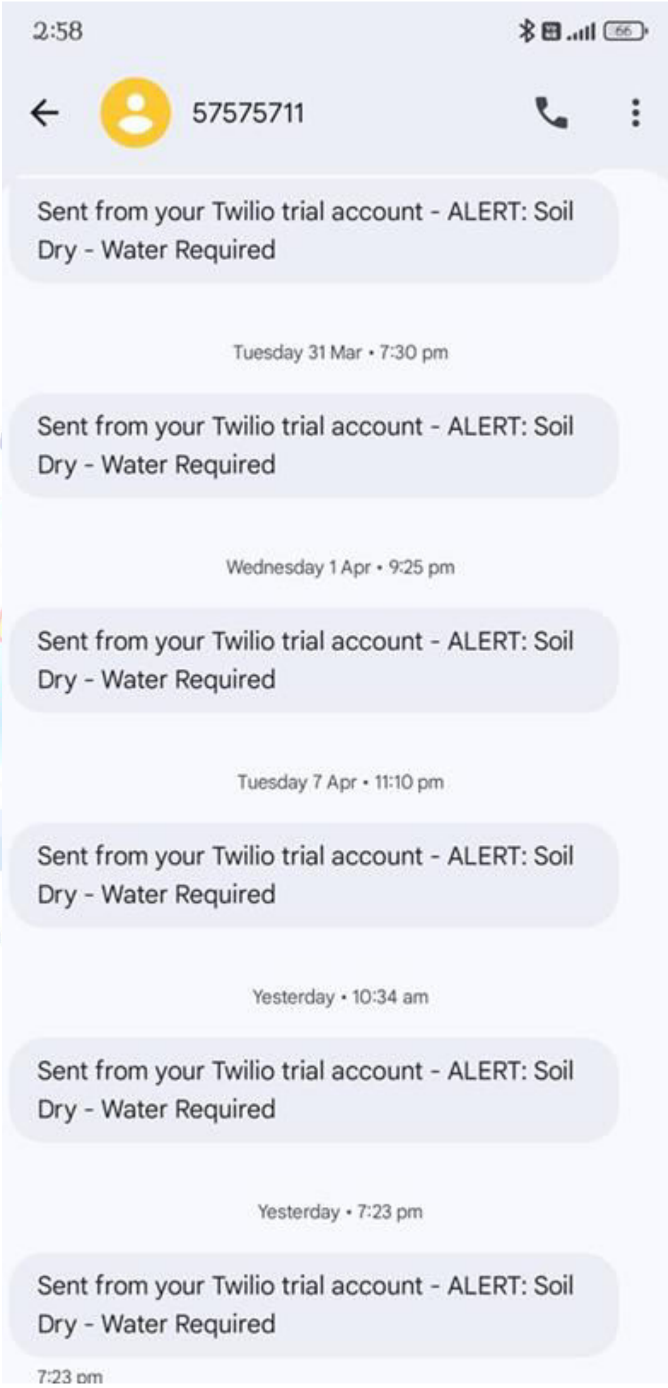

Fig 6: Prediction Result and Analysis

5. CONCLUSION

This project successfully developed an IoT-based climate monitoring system using ESP32 that can measure six environmental parameters, display data locally, send it to the cloud, and generate SMS alerts — all at a low cost of around ₹1,500–₹1,800. The system integrates multiple sensors for temperature, humidity, soil moisture, rainfall, light, and gas detection. Data is shown on an LCD and uploaded to ThingSpeak every 20 seconds for real time monitoring. Alerts are sent via Twilio within 3–5 seconds, along with LED and buzzer indications. Testing over 5 days showed reliable performance with no crashes, stable Wi-Fi reconnection, accurate data uploads, and timely alert delivery without duplicates. Overall, the project proves that a low-cost IoT system can provide efficient, real-time environmental monitoring, making it useful for agriculture, education, and safety applications.

6. FUTURE SCOPE

Several enhancements are planned to extend the system's capability:

• DHT22 Upgrade: Replace DHT11 (GPIO 4, same connection) with DHT22 to improve accuracy to ±0.5°C and ±2% RH and extend range to -40–80°C. Code change: #define DHTTYPE DHT22.

• Calibrated CO2 Sensor: Replace MQ gas sensor with MH-Z19B NDIR sensor (UART interface on GPIO 16/17) for calibrated CO2 concentration in ppm — a direct climate change indicator — instead of the relative gas% currently measured.

• OLED Display: Replace 16×2 LCD with a 128×64 I2C OLED (SSD1306) for simultaneous display of all six parameters and mini trend graphs using the Adafruit SSD1306 library.

MQTT Real-Time Streaming: Replace ThingSpeak HTTP GET with MQTT protocol (PubSubClient library, HiveMQ or AWS IoT broker) to reduce data latency from 20 seconds to under 1 second for true real-time monitoring.

• SD Card Data Logging: Add SD card module (SPI on GPIO 5/18/19/23) to log all sensor readings to CSV files with timestamps, providing local backup during Wi-Fi outages and enabling offline data analysis.

• Solar Power System: Integrate 6V/2W solar panel, TP4056 charging module, and 18650 Li-ion battery for fully autonomous off-grid operation in remote agricultural fields without AC power.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] Postolache, O., Girao, P.S., & Pereira, J.M.D. (2019). Smart Sensors Network for Air Quality Monitoring Applications. IEEE Transactions on Instrumentation and Measurement, 58(9), 3253-3262. https://doi.org/10.1109/TIM.2009.2019196

[2] Marques, G., & Duarte, R. (2017). Indoor Air Quality Assessment Using a CO2, Temperature, and Humidity Sensor. Sensors, 17(5), 1-14. https://doi.org/10.3390/s17051044

[3] Kumar, A., Surendra, A., Mohan, H., Valliappan, K.M., & Kirthika, N. (2018). Internet of Things Based Smart Irrigation using Regression Algorithm. Proceedings ICICCS 2018, IEEE, 67-71. https://doi.org/10.1109/ICCONS.2018.8663044

[4] Farhan, A.A., Pasha, M.A., & Farhan, A. (2020). Comparative Study of IoT Microcontroller Platforms. International Journal of Advanced Computer Science and Applications, 11(3), 83-89.

[5] Saini, A., Bhatt, D., & Dubey, A.K. (2021). IoT-Based Greenhouse Monitoring using ESP8266 and ThingSpeak. International Journal of Engineering Research & Technology, 10(7), 124-130.

[6] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. IEEE Communications Surveys & Tutorials, 17(4), 2347-2376. https://doi.org/10.1109/COMST.2015.2444095

[7] Espressif Systems. (2023). ESP32 Technical Reference Manual v5.1. https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_ manual_en.pdf

[8] Adafruit Industries. (2023). DHT11/DHT22 Sensor Datasheet and Arduino Library. https://learn.adafruit.com/dht

[9] AOSONG Electronics. (2022). DHT11 Product Manual — Temperature and Humidity Module. Guangzhou: AOSONG Electronics Co. Ltd.

[10] MathWorks. (2023). ThingSpeak Documentation — HTTP API Reference. https://www.mathworks.com/help/thingspeak/ 67

[11] Twilio Inc. (2023). Twilio SMS REST API — Messages Resource Documentation. https://www.twilio.com/docs/sms/api/message-resource

[12] Frank, F. de Brabander. (2021). LiquidCrystal_I2C Arduino Library v1.1. GitHub. https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library

[13] Hanwei Electronics. (2022). MQ-2 / MQ-135 Gas Sensor Datasheet. Zhengzhou: Zhengzhou Winsen Electronics Technology Co.

[14] Bhatt, P., & Patel, Y. (2022). IoT-Based Real-Time Environmental Monitoring using ThingSpeak. IJACSA, 13(4), 132-140. Technologies for Homeland Security, 1-6. https://doi.org/10.1109/THS.2016.7568887