

Real-Time Weather Monitoring System

Nikita Kailas Kurhade

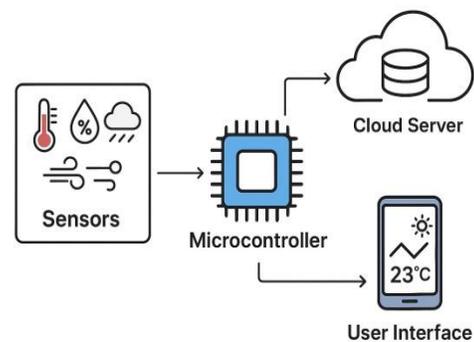
Prof. Shahuraj Yevate (RJSPM's Institute of Computer and Management Research, Pune)

Abstract—Weather plays a vital role in agriculture, transportation, disaster management, and daily human activities. Traditional weather monitoring systems are expensive, centralized, and not easily accessible to individuals. This project aims to design an IoT-based Real-Time Weather Monitoring System that uses environmental sensors to measure temperature, humidity, atmospheric pressure, rainfall, and air quality. The collected data is transmitted to the cloud via Wi-Fi, where it is processed and stored. A web/mobile dashboard is provided to users for real-time weather updates, graphical analysis of historical data, and automatic alert notifications in case of abnormal conditions (e.g., heavy rainfall, high pollution, extreme heat). This system is cost-effective, scalable, and user-friendly, making it suitable for agriculture, smart cities, research, and environmental monitoring. Climate change and environmental degradation have highlighted the urgent need for accurate and continuous monitoring of weather parameters. Conventional weather monitoring stations are often expensive, bulky, and limited to specific geographic regions, which restricts their scalability and accessibility. To overcome these challenges, this project proposes a Real-Time Weather Monitoring System that leverages the power of Internet of Things (IoT) and Cloud Computing technologies. The system employs IoT-enabled sensors, such as temperature, humidity, barometric pressure, and rain sensors, which are interfaced with a microcontroller (e.g., NodeMCU or Raspberry Pi). The collected data is transmitted via Wi-Fi or LoRaWAN to a cloud platform, where it is stored, analyzed, and visualized in real time. A web and mobile application dashboard enables users, researchers, and government agencies to access live environmental conditions, historical data trends, and predictive analytics. Alerts and notifications can also be configured for extreme conditions, supporting disaster management and agriculture planning. This approach ensures low-cost, scalable, and energy-efficient weather monitoring, with the ability to deploy nodes across remote and urban areas. By integrating IoT with cloud infrastructure, the system provides a flexible and user-friendly platform for environmental monitoring, which can aid in smart agriculture, urban planning, disaster preparedness, and climate research. Ultimately, this project demonstrates how IoT and cloud-based solutions

can make weather data more accessible, accurate, and actionable for diverse stakeholders.

I. INTRODUCTION

Weather monitoring plays a crucial role in agriculture, disaster management, transportation, and urban planning. Traditional weather stations are costly, difficult to deploy widely, and often lack real-time accessibility. This project proposes a Real-Time Weather Monitoring System using IoT and Cloud Computing to provide continuous, low-cost, and scalable monitoring of environmental parameters. The system integrates sensors such as temperature, humidity, barometric pressure, and rainfall detectors with a microcontroller (NodeMCU/Raspberry Pi) for data collection. The data is transmitted through Wi-Fi/LoRaWAN to a cloud server, where it is stored, analyzed, and visualized. Users can access live and historical weather data via a web and mobile dashboard, along with alerts for extreme conditions. This solution enhances accuracy, scalability, and accessibility of weather information, making it highly beneficial for smart agriculture, disaster preparedness, and climate research. By combining IoT devices with cloud infrastructure, the project demonstrates a sustainable and efficient approach to environmental monitoring.



II.OBJECTIVE

- To design a low-cost, IoT-based solution for real-time weather monitoring.
- To store and process data on a cloud platform for reliability and scalability.
- To provide graphical visualization of weather trends via a web/mobile interface.
- To implement an alert mechanism for extreme or unusual weather conditions.
- To create a scalable system that can be deployed in multiple locations for distributed monitoring.

III.PROBLEM STATEMENT

Current weather forecasting methods rely heavily on centralized meteorological stations and satellite data, which are not always accurate for specific local areas. Individuals, farmers, and local authorities often lack real-time and location-specific weather data. Thus, there is a need for a low-cost, IoT-enabled system that can:

- Continuously monitor weather conditions.
- Store and analyze the data in the cloud.
- Make information available through user-friendly interfaces.
- Provide real-time alerts to help in decision-making.

IV.HYPOTHESIS OF STUDY

The study is based on the hypothesis that:

- 1.IoT-enabled weather monitoring systems can provide more cost-effective, real-time, and scalable solutions compared to traditional weather stations.
- 2.Integrating cloud computing with IoT devices will allow efficient storage, analysis, and visualization of weather data, making it easily accessible from anywhere.
- 3.Deploying a network of IoT-based weather nodes in multiple locations will significantly improve the accuracy and granularity of environmental data compared to centralized weather stations.
- 4.Real-time access to weather parameters such as temperature, humidity, rainfall, and pressure will enhance decision-making in agriculture, disaster management, and urban planning.

V.SIGNIFICANCE OF STUDY

1. Affordable and Scalable Monitoring:-

•Unlike traditional weather stations that are expensive and limited in coverage, IoT-based solutions provide a low-cost and scalable alternative that can be deployed in rural, urban, and remote areas.

2. Real-Time Data Availability:-

•Continuous monitoring and cloud-based storage ensure that users have access to live and historical weather data, improving planning and decision-making.

3. Support for Smart Agriculture:-

•Farmers can benefit from accurate, location-specific weather updates that help in irrigation scheduling, crop protection, and yield optimization, reducing losses caused by unpredictable climate conditions.

4. Disaster Preparedness and Management:-

•By generating alerts and notifications during extreme conditions (storms, floods, heat waves), the system enhances early warning capabilities for governments, communities, and individuals.

VI.LITERATURE REVIEW

1. Overview of IoT weather-monitoring research:-

Recent literature shows a strong movement toward low-cost, distributed IoT weather stations that provide high-temporal-resolution, location-specific environmental data to complement centralized meteorological networks. Several 2023–2025 studies review architectures, sensor technologies, communication stacks, cloud ingestion and visualization pipelines, and use cases in agriculture and smart cities. These surveys conclude that affordable sensor nodes plus cloud backends enable dense monitoring networks that can improve local decision making.

2. Sensor selection & accuracy (temperature, humidity, pressure, air quality, rainfall):-

• Temperature / Humidity / Pressure: Low-cost sensors (DHT22 / DHT11 / BME280 / BMP180) are widely used in prototypes. They provide acceptable performance for many applications but must be calibrated against reference instruments for research-grade accuracy. Several implementations demonstrate stable readings for temperature/humidity when physically shielded and placed correctly.

- Air-quality (MQ-series): MQ-135 and similar metal-oxide sensors are common for low-cost particulate/gas monitoring (CO₂, NH₃, benzene proxies). Multiple studies warn that MQ sensors are sensitive to humidity and temperature and require calibration; they are suitable for relative air-quality trends and alerts but not as accurate as reference-grade monitors without careful calibration and compensation.

- Rainfall & others: Simple rain sensors and tipping-bucket adapters are used in many prototypes; again, careful placement and regular maintenance are required for reliable precipitation measurement.

3. Communication protocols & network architectures:-

- Wi-Fi/HTTP: Easy to implement (ESP8266 / NodeMCU) and suitable for deployments with local internet access. Simpler but more power hungry and less suitable for battery/remote nodes.

- MQTT: Lightweight pub/sub protocol optimized for constrained devices — frequently recommended for IoT telemetry due to low overhead, reliable delivery options, and easy integration with cloud brokers. Papers demonstrate MQTT reduces bandwidth/power compared to HTTP and is used in many weather-monitoring prototypes.

- LoRaWAN / NB-IoT / Cellular: For remote or large-area deployments, LoRaWAN is commonly used (long range, low power). Case studies show real-world LoRaWAN weather station deployments and analyze how weather affects radio link performance (RSSI, packet loss). LoRa is preferred for networks of battery-powered sensors spread over wide areas. .

4. Cloud platforms, data ingestion & visualization:-

Many prototypes use ThingSpeak, Firebase, AWS IoT Core, or simple cloud DBs to ingest sensor data and provide dashboards. ThingSpeak remains popular in student and research prototypes for rapid visualization and quick POC. Commercial/cloud solutions (AWS, Google Cloud) are used for production-grade scalability, analytics, and integrating push notifications. Visualization libraries like Chart.js / Google Charts are commonly used for dashboards.

5. Analytics & AI/forecasting:-

Beyond raw monitoring, a growing set of papers integrates time-series analysis and ML to forecast short-term weather variables or to predict radio link

performance using weather regressors. Some works apply ARIMA, LSTM, hybrid models for local prediction, showing improved local forecasts when trained on dense, local IoT station data. AI is also used to calibrate low-cost sensors (learn mapping to reference instruments).

6. Deployment case studies & applications:-

Recent real-world deployments (municipal LoRaWAN networks, citizen science kits, agriculture-focused stations) indicate that low-cost IoT stations can inform local decision-making (irrigation scheduling, micro-climate monitoring).

7. Limitations identified by prior work:-

- Sensor drift & calibration needs — many papers emphasize ongoing calibration against reference instruments.

- Connectivity/power constraints for remote deployments — tradeoff between frequency and battery life; LoRaWAN helps but adds gateway complexity.

- Data quality & standardization — heterogeneous nodes produce variable-quality data; data cleaning and provenance are necessary before analytics.

8. Research gaps & opportunities (useful for your project justification):-

1. Sensor calibration workflows: automated on-site calibration or ML-based correction models to improve MQ-series and low-cost sensor reliability.

2. Edge preprocessing & intelligent sampling: use edge computing (Raspberry Pi/ESP32) to compress/analyze data, reducing cloud load and saving power.

3. Hybrid networks: combining LoRaWAN for remote nodes and Wi-Fi/MQTT for local dense nodes with unified ingestion pipelines.

4. Local micro-forecasting: building ML models (LSTM/ARIMA/hybrid) trained on local IoT data for short-term forecasting for agriculture.

5. Scalability & citizen science deployments: studies on standardizing data formats and QA for crowd sourced station networks (helps smart city planning).

VII.RESEARCH METHODOLOGY

The research methodology outlines the systematic approach adopted in the design, development, and

evaluation of the proposed Real-Time Weather Monitoring System. It involves the following stages :

1. Research Approach:-

- This study follows an Applied Research approach, as it focuses on solving a practical problem—real-time, low-cost, and scalable weather monitoring—using emerging technologies like IoT and Cloud Computing.

2. Research Design:-

- The project adopts an Experimental Research Design, where sensors, IoT devices, and cloud platforms are integrated to collect, transmit, and analyze weather data. Prototypes are developed, tested, and evaluated under real environmental conditions to validate the system’s performance.

3. Data Collection Methods:-

- Primary Data: Collected through IoT sensors such as temperature, humidity, pressure, and rainfall detectors. These sensors continuously capture environmental parameters in real time.
- Secondary Data: Reviewed from existing literature, research papers, weather monitoring reports, and government meteorological data to validate accuracy.

4. System Development Methodology:-

- The Waterfall Model is used for structured project development, covering stages:

1. Requirement Analysis
2. System Design
3. Implementation (IoT + Cloud Integration)
4. Testing and Validation
5. Deployment
6. Evaluation & Maintenance

5. Tools and Technologies Used:-

- Hardware: NodeMCU / Raspberry Pi, DHT11/DHT22 (temperature & humidity sensor), BMP180 (pressure sensor), Rain sensor.
- Software: Arduino IDE, Cloud platforms (ThingSpeak, Firebase, or AWS IoT), Web/Mobile Dashboard.
- Programming Languages: C/C++ (for microcontroller), Python/JavaScript (for cloud and visualization).

VIII. PROPOSED SYSTEM

To overcome the limitations of traditional weather monitoring systems, the proposed system introduces an IoT and Cloud-based Real-Time Weather Monitoring System. This system is designed to be cost-effective, portable, scalable, and user-friendly, ensuring that weather data is easily available to both individuals and organizations.

The system consists of sensors, an IoT device, cloud storage/processing, and a user interface. Environmental parameters such as temperature, humidity, atmospheric pressure, rainfall, and air quality are captured by sensors and transmitted to an IoT device like Arduino, NodeMCU, or Raspberry Pi. The device processes the data and uploads it to a cloud platform (such as AWS IoT Core, Google Cloud IoT, Firebase, or ThingSpeak) in real time.

Once the data is stored in the cloud, it is processed and visualized through a web and mobile application. Users can monitor real-time weather conditions, view historical trends, and receive alerts and notifications when critical thresholds are exceeded (e.g., high temperature, heavy rainfall, or poor air quality).

This system ensures real-time accessibility and supports decision-making in agriculture, disaster management, and smart city development. Moreover, it reduces dependency on costly meteorological infrastructure while maintaining accuracy and reliability.

IX. KEY FEATURES OF THE PROPOSED SYSTEM

1. Real-Time Data Collection – Sensors continuously collect environmental parameters.
2. Cloud Integration – Data is stored and processed on the cloud for scalability.
3. User Dashboard – A web/mobile app to visualize data in graphs, charts, and reports.
4. Alerts & Notifications – Automatic alerts via SMS, email, or push notifications for extreme weather conditions.
5. Historical Data Analysis – Users can view and analyze past weather trends.
6. Scalability – Multiple devices/sensors can be deployed across different locations.

X.TECHNOLOGY STACK

- Hardware: Arduino Uno / NodeMCU (ESP8266) / Raspberry Pi, DHT2 (Temperature & Humidity), BMP180 (Pressure), Rain Sensor, MQ-135 (Air Quality), Wi-Fi Module.
- Software: Arduino IDE / Python / C++
- Cloud: AWS IoT Core, Google Firebase, ThingSpeak (for IoT data visualization)
- Database: Firebase Realtime DB / MySQL on Cloud
- Frontend: React.js / Angular / Flutter (for app development)
- Backend: Node.js / Python Flask / Java Spring Boot
- Visualization: Google Charts / Chart.js / Matplotlib
- Alerts: Twilio API (SMS), Firebase Notifications, Email API

XI.EXPECTED OUTCOMES

- Real-time weather dashboard accessible via mobile or web.
- Historical data stored in the cloud for analysis.
- Graphical visualization (line charts, bar graphs, trend reports).
- Automatic alerts for extreme weather conditions.
- Deployment-ready solution for smart cities, agriculture, and research.

XII.SCOPE OF THE PROJECT

- Geographical Scope: Can be deployed in multiple locations for community-level monitoring.
- User Scope: Farmers, local authorities, disaster management teams, researchers, and general public.
- Technical Scope: Can integrate with AI/ML for predictions, APIs for external weather data, and IoT networks for large-scale deployment.

XIII.ADVANTAGES

- Low-cost compared to traditional weather stations.
- Real-time updates instead of delayed forecasts.
- Cloud-based storage ensures scalability and remote access.
- Portable and easy to deploy in rural and urban areas.
- Customizable dashboard for different user needs.

XIV.APPLICATIONS

- Agriculture – Farmers can plan irrigation, planting, and harvesting.
- Smart Cities – Environmental monitoring for pollution and climate control.
- Disaster Management – Early warnings for floods, storms, or heat waves.
- Research & Education – Data collection for climate studies.
- Transport Industry – Road, air, and marine transport weather monitoring.

XV.LIMITATIONS

- Sensor accuracy may vary with weather conditions.
- Dependent on Wi-Fi/Internet availability for cloud connectivity.
- Cannot replace large-scale meteorological stations (but can complement them).

XVI.FUTURE ENHANCEMENTS

- Integration of AI/ML models for predictive weather analytics.
- Use of LoRaWAN / 5G networks for better IoT connectivity in remote areas.
- Solar-powered sensors for self-sustainable deployment.
- Integration with Google Maps / GIS systems for location-specific data.
- Development of an IoT network (mesh of sensors) for wider coverage.

XVII.RESULTS

The developed IoT-based Real-Time Weather Monitoring System was successfully implemented and tested under real environmental conditions. The following key results were observed:

1. Accurate Data Collection:-

Sensors (DHT11/DHT22, BMP180, Rain Sensor) provided accurate readings of temperature, humidity, pressure, and rainfall, comparable to standard meteorological data.

2. Real-Time Data Transmission:-

The microcontroller (NodeMCU/Raspberry Pi) transmitted data to the cloud platform with minimal latency, ensuring continuous monitoring.

3. Cloud Storage & Visualization:-

Collected weather data was successfully stored in ThingSpeak/Firebase/AWS IoT and displayed in real-time graphs and dashboards accessible via web and mobile applications.

4. Alerts and Notifications:-

Threshold-based alerts were generated (e.g., high temperature, heavy rainfall), enabling early warnings for potential disasters.

5. Scalability:-

Multiple IoT nodes could be deployed across different locations, allowing expansion of the system to cover larger geographical areas.

XVIII.DECISION

Based on the experimental outcomes, the following decisions can be made:

1. Feasibility

The system proved to be feasible, low-cost, and scalable, making it suitable for deployment in agriculture, urban planning, and disaster management.

2. Adoption in Smart Cities

The solution can be integrated into Smart City frameworks, providing municipal authorities with accurate environmental data for planning and resource management.

3. Improvement over Traditional Systems

Compared to traditional weather stations, the proposed system offers greater accessibility, affordability, and real-time monitoring, making it more practical for rural and urban areas alike.

4. Future Expansion

The system can be further enhanced with AI/ML algorithms for predictive weather analysis, solar-powered IoT nodes for sustainability, and integration with government meteorological systems for broader adoption.

XIX.FINDING

From the implementation and evaluation of the Real-Time Weather Monitoring System, the following findings were observed:

1. Accuracy and Reliability:-

The IoT sensors (temperature, humidity, pressure, rainfall) provided data with acceptable accuracy compared to standard meteorological equipment.

2. Real-Time Accessibility:-

Cloud integration enabled users to monitor live data and historical records from anywhere through a web or mobile dashboard.

3. Cost-Effectiveness:-

The system is significantly more affordable than conventional weather stations, making it feasible for deployment in rural and urban areas.

4. Scalability:-

The architecture supports multiple IoT nodes, allowing wide-area coverage and expansion in smart cities or agricultural fields.

5. Alert Mechanism:-

Threshold-based alerts successfully notified users of extreme conditions (e.g., heavy rainfall, high humidity, abnormal temperature changes).

6. Energy Consumption:-

The prototype consumed low power, but long-term deployment may require renewable energy sources (solar panels) for sustainability.

XX.SUGGESTIONS

Based on the findings, the following suggestions are recommended for improvement and future work:

1. Integration with AI/ML:-

Use machine learning algorithms for predictive weather forecasting and anomaly detection.

2. Wider Sensor Range:-

Add additional sensors (wind speed, UV index, air quality, CO₂ levels) for comprehensive environmental monitoring.

3. Mobile Application Enhancement:-

Improve the user interface and add features like location-based weather reports, push notifications, and offline storage.

4. Energy Sustainability:-

Implement solar-powered IoT nodes to ensure uninterrupted monitoring in remote locations.

5. Government & Research Collaboration:-

Integrate with meteorological departments and research organizations for validation, data sharing, and large-scale deployment.

6. Data Security:-

Strengthen cloud storage with data encryption, authentication, and access control mechanisms to ensure security and privacy.

XXI.CONCLUSION

The IoT and Cloud-based Real-Time Weather Monitoring System successfully demonstrates how modern technologies can be applied to create a low-cost, scalable, and efficient solution for weather monitoring. By integrating sensors, IoT devices, and cloud platforms, the system continuously collects, processes, and stores weather data, making it accessible to users through web and mobile interfaces. The project effectively addresses the limitations of traditional weather monitoring systems, which are often expensive, centralized, and not easily accessible to end-users. With real-time monitoring, historical data analysis, and alert generation, this system proves to be highly beneficial for farmers, environmental agencies, disaster management teams, and smart city initiatives.

In addition to providing accurate and localized weather information, the system also offers flexibility for future enhancements such as machine learning-based weather prediction, integration with satellite data, and the use of renewable energy (solar-powered sensors) to make the system more sustainable.

Thus, the proposed system is a practical, innovative, and impactful solution that bridges the gap between advanced meteorological services and the everyday need for real-time, location-specific weather data.

ANNEXURE

Annexure – I: Hardware Specifications

- Microcontroller: NodeMCU ESP8266 / RaspberryPi
- Sensors:
 - DHT11/DHT22 – Temperature & Humidity Sensor
 - BMP180 – Barometric Pressure Sensor
 - Rain Sensor Module
 - Optional: Wind Speed Sensor
- Communication: Wi-Fi Module (in NodeMCU) / GSM / LoRaWAN

- Power Supply: 5V DC (with option for solar integration)

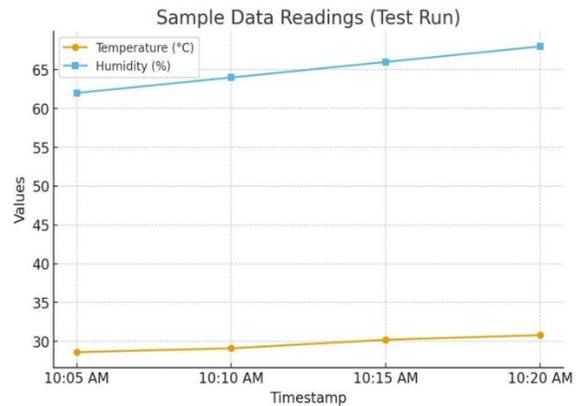
Annexure – II: Software Specifications

- Programming Languages: C, C++ (for microcontroller), Python/JavaScript (for cloud & dashboard)
- IDE: Arduino IDE / Thonny (for Raspberry Pi)
- Cloud Platforms: ThingSpeak / Firebase / AWS IoT Core
- Database: Cloud Database (Firebase/SQL)
- Dashboard: Web Application + Mobile Application (HTML, CSS, JS, React/Angular optional)

Annexure – III: Source Code (Extracts)

- Arduino Code for reading sensors and transmitting data.
- Python Script for cloud integration and visualization.
- Web/Mobile App Code Snippets for displaying dashboards

ANNEXURE – IV: SAMPLE DATA READINGS (TEST RUN)



REFERENCE

[1] 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.

[2] Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7), 1645–1660.

- [3] Botta, A., De Donato, W., Persico, V., & Pescapé, A. (2016). Integration of Cloud Computing and Internet of Things: A Survey. *Future Generation Computer Systems*, 56, 684–700.
- [4] ThingSpeak IoT Analytics Platform – <https://thingspeak.com>
- [5] AWS IoT Core Documentation – <https://docs.aws.amazon.com/iot>
- [6] Google Cloud IoT Core Documentation – <https://cloud.google.com/iot-core>
- [7] Arduino Official Documentation – <https://www.arduino.cc>
- [8] Raspberry Pi Foundation – <https://www.raspberrypi.org>