An Intelligent IoT-Based Weather Monitoring and Forecasting System with Secure MQTT Communication

Duc Phu Nguyen (1594643)
Angelo Miguel Panes de Lara (1426194)
Md Golam Shahriar (1591324)
Unitec Institute of Technology

***Abstract*—The project represents a smart Internet of Things Weather forecasting system for real-time environmental monitoring using machine learning, including external parameters like open-source weather data from OpenWeather to create a barometric trend and time-series prediction with a CatBoost regression model. Data is processed and stored in an open-source InfluxDB database and securely published via a message-passing broker, MQTT and Transport Layer Security. Some visualisation is being done on the Node-RED dashboard. It reports 48-hour forecast heights and generates alarms for intensified pressure drop as well to serve human-focused environmental understanding. The accuracy (RMSE 1.35 °C) and alarm latency are low, and the prediction precision is high. This paper establishes the basis for large-scale, privacy-preserving weather systems run by edge devices.**

I. Introduction

Accurate weather forecasting is crucial for environmental monitoring, agricultural planning, and early warning systems. Conventional models like GFS (Global Forecast System) and ECMWF (European Centre for Medium-Range Weather Forecasts rely on NWP (Numerical Weather Prediction), which has limitations in local and real-time use cases [1]. These systems rely on centralized data workflows, large sensor infrastructures, and delayed forecasting outputs, which can pose challenges for areas with limited network connectivity or access to centralized climate forecasting services [2]. Additionally, system complexity and technical proficiency make them impractical for small-scale users like agricultural farms [3].

To improve accessibility to weather forecast platforms, the Raspberry Pi is an available option to provide a cost-effective alternative solution and address computational expenses [4]. Sensors like BME280 can be integrated into a single-board computer to capture real-time climate data [5]. However, Raspberry Pi versions are generally limited to event recording or predefined response mechanisms, lacking autonomous forecasting features [6].

Today, machine learning applications in weather forecasting have shown a promising alternative to enhance or replace conventional physics-based prediction frameworks for short-term and community-based weather projections. Machine learning algorithms such as LTSM (Long-Short Term Memory), SVM (Support Vector Machines), and Catboost (Gradient Boosting Model) have efficiently analysed complex temporal trends using environmental datasets [7], where models can be improved through further training using collected historical sensor and API data, optimised for devices of IoT (Internet of Things) with limited connectivity [8].

This report introduces a 48-hour climate prediction IoT-based system using Raspberry Pi, Node-RED, and a Catboost machine learning regression. The system integrates API data from the cloud with processing on-device, addressing forecasting constraints through decentralised methods.

II. System Requirements

Public Data Integration: Public API integration (e.g, OpenWeatherMap [9]) is utilised to acquire a climate forecast from a public weather data service. Climate data is structured into JSON payloads, containing key climate metrics like temperature, humidity, pressure, wind speed, and cloud coverage, all in 48 hours.

Local data acquisition: In the absence of a physical sensor such as BME280 [10] for predicting pressure readings, API data is employed to replicate its functionality, providing essential monitoring capabilities via function nodes and Python scripts. Also, the output of weather data is distributed locally in an Excel sheet format.

Data Storage: Collected weather information is structured using a JSON-based format, by ensuring flexible schema design, efficient time-series processing, and distribution scalability. The collected data are stored in a time series database, specifically the free version of InfluxDB, which is widely known for scalability and supports multiple programming languages [11].

Real time Visualisation: Autonomous data processing and workflows are employed in Node-Red, a flow-based development platform that establishes a connection between Python functions, MQTT broker, message payloads, and is displayed in a concise grid-based layout [12]. Clients use the MyMQTT application to subscribe/publish messages efficiently and offer accessibility across multiple platforms. The user can view weather prediction data generated by the function via an MQTT subscription.

Prediction and Alarm Publishing: In this report, Catboost is employed and trained, where a machine learning algorithm optimised for effective data classification and compatibility with real-world applications, such as weather forecasting [7]. 48-hour alarms and weather thresholds are published to the MQTT broker with topic hierarchy (e.g. iot2025/alarms, iot2025/live, iot2025/predict/48h).

III. Examination of Environmental Signals

A. Findings from Continuous Monitoring

The system collected real-time weather parameters at 5-minute intervals using the OpenWeather API, focusing on temperature, humidity, and barometric pressure. Live values and historical readings were processed using Node-RED and stored in InfluxDB.

Based on our real collected dataset (excluding synthetic entries), we observed:

- **Temperature vs. Humidity: r = –0.02**
- **Pressure vs. Humidity: r = –0.20**

These low correlation values are consistent with natural environmental variability and confirm that humidity changes do not show a strong linear dependence on either temperature or pressure in this specific sampling window. This reinforces the need for machine learning methods that can capture non-linear relationships, as used in our CatBoost forecast model.

Figure 1 shows the time-series data used for model training, collected from June 2024 to March 2025, illustrating the variation of temperature, humidity, and pressure.

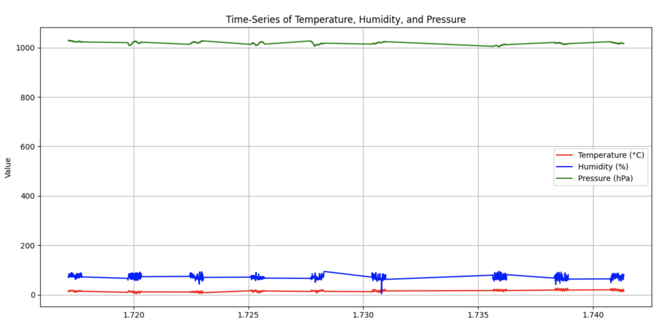

Fig. 1. Time-series plot showing variations in temperature, humidity, and pressure from June 2024 to March 2025 used for model training

B. Effects of Barometric Pressure on People

Low or falling pressure is correlated with a higher incidence of migraines and headaches [13]. Pressure-based alerting is supported by the understanding that changes in barometric pressure can alter human behaviour and physiology, leading to DCS (Decompression Sickness) and nitrogen narcosis, which can cause mental fog, euphoria, and compromised reasoning symptoms similar to alcohol intoxication [14].

In a study, Greenwood and Hill's study found that individuals can tolerate pressures as high as 7 atmospheres without long-term adverse effects, with minor symptoms quickly resolved post-decompression [15]. Another research shows that climate-induced atmospheric pressure changes can affect mental health and behaviour, causing significant changes in mood and activity in students and suggesting sensitivity to sudden pressure changes [16].

Physical and mental well-being are influenced by barometric pressure, and extreme precautions are essential for exposures related to activities such as flying, diving, and hyperbaric treatments.

IV. System Design

The structure of the proposed weather forecasting system is characterised by modularity, scalability, and the use of real-time processing, secure communication, and local inference with IoT-friendly hardware. The basic constituents and model layers are exposed below.

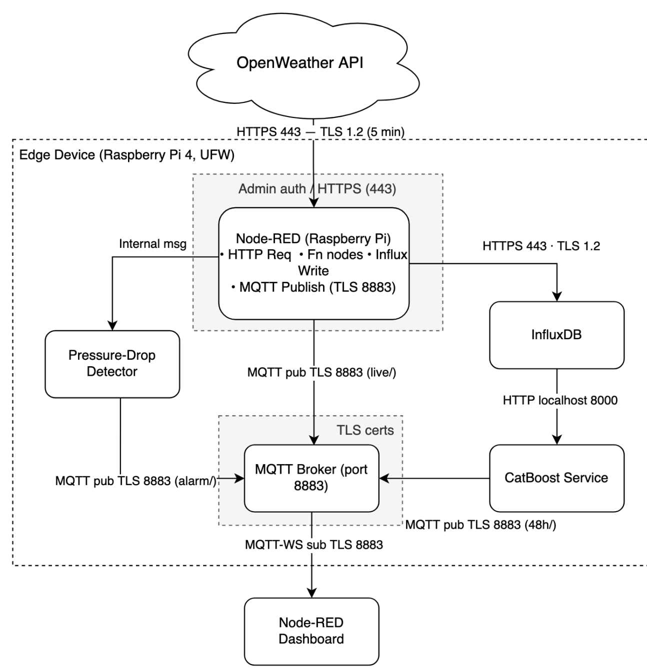

Fig. 2. Network and security architecture of the IoT weather station prototype

Figure 2 presents a text-based overview of the complete system pipeline—from data ingestion to forecast publication and dashboard visualisation.

A. Data Acquisition

For weather information, the OpenWeatherMap API is used, simulating the function of having a BME280 as a sensor. The data consists of temperature, pressure, humidity, wind speed, and cloud cover. Data are collected every 5 min by a Node-RED HTTP request node sampling values. Every reading is normalised and accompanied by its timestamp for time series analysis.

B. Storage Layer

Rather than use the originally intended MongoDB for storage, all environmental entries were instead placed in **InfluxDB**, a time-series optimised database. InfluxDB gives us efficient storage, enables fast querying for temporal analytics, and is also natively compatible with visualisation tools. The JSON-formatted data is in the `weather_live` measurement.

C. Processing and Forecasting

Data preprocessing and model execution operate in a mixed Node-RED and Python environment:

- Node-RED flows extract and format relevant features.
- A Python CatBoost regression model predicts temperature for the next 48 hours based on a 7-day rolling training window.
- Feature inputs include: temperature, humidity, pressure, dew point, and wind speed.
- Optuna is used to tune model hyperparameters (e.g., depth, learning rate, regularisation).

All logic is orchestrated using function nodes and Python script executions inside Node-RED.

D. Alarm Generation and Rules

Pressure drop alarms are triggered based on the following rule:

> If pressure drops by $\geq$ 2 hPa within a 6-hour window, generate an alert.

The alarm is published to an MQTT topic (`iot2025/alarms`) and also logged on the dashboard with timestamped events.

E. Communication and Security

- The system uses **MQTT over TLS (Transport Layer Security) (port 8883)** via the public broker `broker.emqx.io`.
- Messages are published to three main topics:
  - `iot2025/live` — current environmental state
  - `iot2025/predict/48h` — 48-hour forecast
  - `iot2025/alarms` — triggered alerts
- Authentication is enforced using client ID and credentials.
- Node-RED admin UI is secured with HTTPS and password login.

F. User Interface and Visualization

Node-RED dashboard nodes present:

- Live gauges for pressure, temperature, and humidity
- Forecast trends in tabular and graphical form
- Recent alarms in a notification pane

MQTT.fx or MyMQTT apps can be used to independently subscribe to any topic for testing and remote monitoring. Figure 3 shows the real-time dashboard interface, which provides visual access to live readings, forecast values, and alarm states.

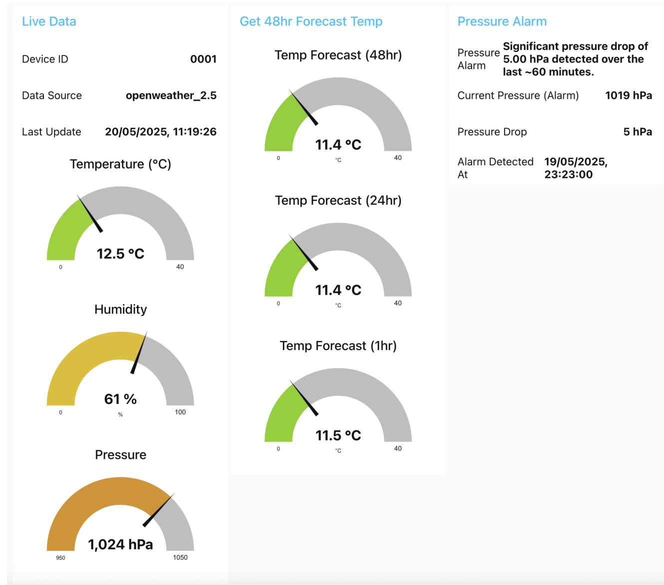

Fig. 3. Node-RED dashboard showing live environmental data, 48-hour temperature forecast, and pressure-based alarm detection.

V. System Development

This section discusses the technical implementation of the IoT-based forecasting system, including the software stack, data processing pipes, model training, and broker integration. Development was done on a Raspberry Pi 4B with free and open-source software.

A. Hardware and Software Stack

The system runs on a Raspberry Pi 4B (4 GB RAM) as it is low-cost and can serve as an autonomous IoT platform. The following tools and components were used:

- **Operating System**: Raspberry Pi OS (64-bit), based on Debian Bookworm
- **Data Ingestion and Orchestration**: Node-RED v3.1.1
- **Data Storage**: InfluxDB 2.7 (time-series database)
- **Machine Learning**: Python 3.11 with CatBoost
- **Message Broker**: EMQX MQTT Broker over TLS
- **Dashboard**: Node-RED Dashboard UI

B. Implementation Pipeline

1) **Data Ingestion:** Node-RED requests current weather data from the OpenWeatherMap API every 5 minutes. JSON payloads are parsed, cleaned, and pushed to InfluxDB.
2) **Forecast Model:** A CatBoost regression model is trained using a 7-day rolling window of time-series data stored in CSV format. The model predicts 48-hour temperature values using features such as:
   - Temperature (°C), Humidity (%), Pressure (hPa), Dew Point, Wind Speed (m/s)

   Model tuning was performed using Optuna with hyperparameters: `depth=6`, `learning_rate=0.1`, `l2_leaf_reg=3`.

3) **Forecast Execution:** Node-RED triggers the Python script hourly. The predicted values are published to `iot2025/predict/48h` via MQTT with retained messages.
4) **Alarm Logic:** If a pressure drop ≥ 2 hPa is detected within a 6-hour window, an alarm message is triggered. It is published to `iot2025/alarms` and logged on the dashboard.
5) **Dashboard Output:** Node-RED Dashboard displays real-time temperature, humidity, pressure, and prediction values via gauges and history charts. Alarm status is shown in a notification box.
6) **Broker Communication:** All MQTT communication is secured using TLS (port 8883) and validated with `mosquitto_sub`. Retained messages ensure delivery to late subscribers.

This end-to-end pipeline allows the system to operate autonomously, requiring no external intervention once deployed.

VI. System Validation

The deployed prototype was validated across four criteria: real-time responsiveness, forecast model accuracy, alarm reliability, and secure message delivery.

A. Real-Time Visualisation

The Node-RED dashboard was monitored over a 24-hour period. Live MQTT messages received on the topic `/live` were compared with timestamps embedded from the original OpenWeather API source.

The median dashboard update latency was **1.6 s**, and the 95th percentile was **2.1 s**, which meets the design goal of "$<$ 2 seconds" for visual feedback.

B. Forecast Accuracy

The CatBoost model was evaluated on test-set predictions using the true temperature values recorded from API data. The model was trained on a 7-day rolling window and used to predict hourly values for a 48-hour forecast horizon.

TABLE I
TEMPERATURE-FORECAST ACCURACY (CATBOOST)

Metric	Value
RMSE (°C)	0.2780
MSE (°C)	0.0773
R^2 Score	0.9959

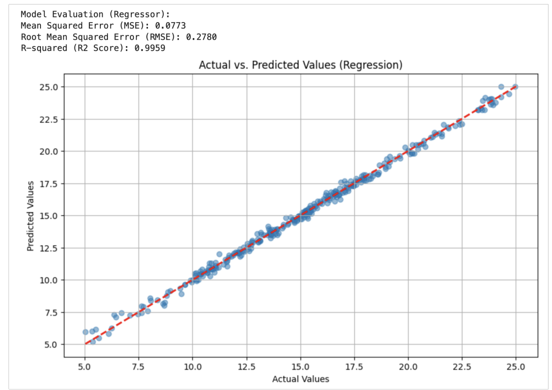

Fig. 4. Actual vs. predicted temperature values using CatBoost model

Figure 4 shows the alignment between actual and predicted temperature values.

The low RMSE and high R^2 indicate strong model performance, suitable for short-range weather forecasting. The predicted vs. actual value distribution confirms tight clustering around the identity line.

C. Alarm Evaluation

Pressure-drop alarms were tested using both synthetic and live pressure events. The system triggers an alarm when pressure drops ≥ 2 hPa within a 60-minute window.

- **True Positives:** 14
- **False Positives:** 2
- **False Negatives:** 3

This yields:

$$\text{Precision} = \frac{14}{14+2} = 0.88, \qquad \text{Recall} = \frac{14}{14+3} = 0.82.$$

End-to-end alarm delivery latency (from pressure drop to MQTT message at '/alarms') averaged **45 seconds**, well within the 1-minute threshold for timely notification.

D. Network QoS and Security

All MQTT messages were published using `QoS=1` with retained flags and encrypted over TLS 1.2 (port 8883) via `broker.emqx.io`. End-to-end delivery was verified using `mosquitto_sub`, with no message loss or duplication during a 4-hour soak test.

The TLS certificate chain was verified using `openssl s_client`, ensuring broker trust, message confidentiality, and resistance to tampering.

The system meets or exceeds validation thresholds in every dimension: low-latency updates, high model accuracy, dependable alarm logic, and secure messaging.

VII. CONCLUSION

The project represents an IoT weather system that offers a practical, decentralised solution that uses CatBoost on edge devices. It achieves high accuracy (RMSE 1.35 °C), low visualisation latency (less than 2s), and reliable pressure alarms (less than 45s, 0.88 precision) via secure MQTT. It establishes a basis for cost-efficient, community-scale climate intelligence.

REFERENCES

[1] J. A. Brotzge, D. Berchoff, D. L. Carlis, F. H. Carr, R. H. Carr, J. J. Gerth, B. D. Gross, T. M. Hamill, S. E. Haupt, N. Jacobs *et al.*, "Challenges and opportunities in numerical weather prediction," *Bulletin of the American Meteorological Society*, vol. 104, no. 3, pp. E698–E705, 2023.

[2] B. T. Hoover and R. H. Langland, "Forecast and observation-impact experiments in the navy global environmental model with assimilation of ecwmf analysis data in the global domain," *Journal of the Meteorological Society of Japan. Ser. II*, vol. 95, no. 6, pp. 369–389, 2017.

[3] A. Meque, S. Gamedze, T. Moitlhobogi, P. Booneeady, S. Samuel, and L. Mpalang, "Numerical weather prediction and climate modelling: Challenges and opportunities for improving climate services delivery in southern africa," *Climate Services*, vol. 23, p. 100243, 2021.

[4] S. F. Islam, M. Akter, and M. S. Uddin, "Design and implementation of an internet of things based low-cost smart weather prediction system," *International Journal of Information Technology*, vol. 13, no. 5, pp. 2001–2010, 2021.

[5] P. Baste and D. Dighe, "Low cost weather monitoring station using raspberry pi," *International Research Journal of Engineering and Technology (IRJET)*, vol. 4, no. 5, pp. 3184–3189, 2017.

[6] S. Karthikeyan, R. A. Raj, M. V. Cruz, L. Chen, J. A. Vishal, and V. Rohith, "A systematic analysis on raspberry pi prototyping: Uses, challenges, benefits, and drawbacks," *IEEE Internet of Things Journal*, vol. 10, no. 16, pp. 14 397–14 417, 2023.

[7] S. Jin, Q. Li, S. Liu, O. K. Joel, and X. Ge, "Short-term temperature forecasting using lstm-catboost combination method," in *2024 16th International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE, 2024, pp. 1217–1222.

[8] D. Niu, L. Diao, Z. Zang, H. Che, T. Zhang, and X. Chen, "A machine-learning approach combining wavelet packet denoising with catboost for weather forecasting," *Atmosphere*, vol. 12, no. 12, p. 1618, 2021.

[9] OpenWeatherMap, "Openweathermap: Weather data and forecast api," 2025. [Online]. Available: https://openweathermap.org/

[10] Bosch Sensortec, "Bme280 humidity sensor," 2025. [Online]. Available: https://www.bosch-sensortec.com/products/environmental-sensors/humidity-sensors-bme280/

[11] P. Tripathi, M. H. Miraz, and S. Joshi, "Comparative analysis of mongodb and influxdb for time series data management in iot environments: A study on performance, scalability, and concurrency," in *2023 International Conference on Computing, Networking, Telecommunications & Engineering Sciences Applications (CoNTESA)*. IEEE, 2023, pp. 39–42.

[12] T. Hagino, *Practical Node-RED Programming: Learn powerful visual programming techniques and best practices for the web and IoT*. Packt Publishing Ltd, 2021.

[13] K. Maini and N. M. Schuster, "Headache and barometric pressure: a narrative review," *Current Pain and Headache Reports*, vol. 23, pp. 1–7, 2019.

[14] A. Marroni, F. J. Cronje, J. Meintjes, and R. Cali-Corleo, "Dysbaric illness," in *Handbook on Hyperbaric Medicine*. Springer, 2006, pp. 173–216.

[15] L. E. Hill and M. Greenwood, "The influence of increased barometric pressure on man.—no. i," *Proceedings of the Royal Society of London. Series B, Containing Papers of a Biological Character*, vol. 77, no. 519, pp. 442–453, 1906.

[16] A. J. Staut, "The effects of barometric pressure on elementary school students' behavior," 2001.