Weather Forecasting Temperature and Rain Using BME280

*Note: Sub-titles are not captured in Xplore and should not be used

Anh Nhat Thai
Bachelor of Data Science
Deakin University
Avondale Heights, 3034
Email: s223339662@deakin.edu.au

***Abstract*— This study investigates the creation of a personal weather forecasting system that predicts short-term weather conditions, such as temperature variations and the chance of rain, utilizing the BME280 sensor and Arduino Nano 33 IoT. The temperature, humidity, and pressure readings from the BME280 sensor are sent to cloud storage for further processing and display via an interactive Plotly Dash interface. For temperature forecasting, a Linear Regression model is used, and for rain prediction, a rule-based approach. With dual storage guaranteeing data stability and real-time warnings enhancing user experience, the system successfully illustrated how inexpensive hardware can deliver localized weather information. This method provides a useful, approachable tool for individual environmental monitoring and forecasting, notwithstanding the simplicity of the forecasting models.**

Keywords—Plotly Dash, temperature, pressure, humidity, Arduino Nano 33 IoT, BME280, Linear Regression

I. Introduction

Weather forecasting is essential in daily life, aiding individuals in planning activities and making informed decisions. Recent advancements in the Internet of Things (IoT) have enabled the development of affordable, localized weather stations that provide personalized insights. This research, conducted by Anh Nhat Thai presents a weather forecasting system using low-cost components: the BME280 sensor from Bosch Sensortec and Arduino Nano 33 IoT. The BME280 sensor, produced by Bosch Sensortec, accurately measures temperature, humidity, and pressure, while the Arduino enables wireless data transmission and cloud integration. The system stores data both locally and in the cloud for reliability and accessibility. Temperature forecasting is conducted using a Linear Regression model, and rain prediction is implemented using a rule-based approach based on humidity and pressure thresholds. An interactive Plotly Dash dashboard visualizes the data, allowing users to explore weather trends. This study demonstrates that even with basic hardware, a practical and reliable localized weather forecasting system can be developed, showcasing the potential of IoT technology to provide accessible environmental insights.

II. Literature review

A. Weather Forecasting System

For many years, weather forecasting has been an integral part of daily life, offering crucial information for transportation, agriculture, and other daily operations. Conventional weather stations forecast variables like temperature, air pressure, and rainfall using sophisticated meteorological equipment and intricate algorithms. But because to technological developments, especially the emergence of the Internet of Things (IoT), it is now feasible to create personal weather monitoring devices that are smaller, easier to use, and more reasonably priced [1] .The typical user may now access personalised weather stations thanks to IoT-based systems made of sensors and microcontrollers that can gather, analyse, and transmit data in real-time. These developments provide consumers more control over hyper-local weather information, facilitating well-informed decision-making in day-to-day activities.

B. Weather Prediction's Sensor

In IoT-driven systems for tracking weather, the selection of a sensor significantly impacts the reliability and exactness of the collected information. The BME280 sensor, manufactured by Bosch Sensortec, stands out as a leading option for environmental tracking. It excels in measuring temperature, humidity, and air pressure, rendering it perfect for applications in weather tracking initiatives [3]. The BME280 has a solid history of dependability and accuracy in monitoring environmental shifts, earning it a high demand among amateurs and scientists [5]. By connecting the BME280 to a microcontroller such as the Arduino Nano 33 IoT, there's an effective method for gathering and sending weather data. The Arduino Nano 33 IoT features an in-built WiFi module, which is ideal for cloud-based data storage and processing, crucial for long-distance monitoring and visualization [2]. The use of these affordable components simplifies the creation of personal weather tracking devices, making them accessible and practical for a broad spectrum of users.

C. Forecasting Techniques

In this initiative, predicting the weather's temperature is achieved through the application of a Linear Regression algorithm. Linear Regression stands as a straightforward yet effective statistical method for identifying temperature trends across periods, making it an appropriate choice for a basic examination of climate patterns [4]. Despite its inability to recognize intricate, non-linear associations, its clarity lends it an advantage in this scenario, where the main objective is to offer insights that are both comprehensible and useful. Determining if it will rain is carried out with a rule-based system, which sets limits on humidity and air pressure to gauge the chance of precipitation. The limits for these criteria are derived from real-world data: elevated humidity (above 80%) along with a decrease in air pressure typically signals a higher chance of rain. This method is simple yet robust for making short-term forecasts and offers definitive signals about possible shifts in weather conditions [1]. Although more advanced models, such as neural networks, might enhance precision, the rule-based system's simplicity makes it accessible and understandable for individual use.

III. Methods

The system for predicting the weather is constructed with affordable electronic parts, mainly the BME280 sensor and the Arduino Nano 33 IoT. The BME280 sensor, made by Bosch Sensortec, accurately gauges temperature, humidity, and air pressure [3]. The Arduino Nano 33 IoT is employed to gather information from the BME280 sensor and send the information to the cloud through its WiFi features, offering an easy way to monitor data remotely and store it in the cloud [1].

A. Data Collection

Data Collection Frequency: Information was gathered on a minute-by-minute basis to accurately monitor weather changes without burdening the system with too much data. This approach allowed for detailed analysis and predictions over short periods. The information gathered by the Arduino was stored on an SD card, acting as a local safeguard to prevent data loss in the case of problems with internet access. Additionally, the data was uploaded to a cloud service, making it available from anywhere. Employing cloud storage facilitated the creation of real-time visualizations on an interactive dashboard [2].

B. Data Visualization

```
# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train the Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)
```

Weather Forecasting: A Straightforward Linear Regression model was applied to estimate changes in temperature from past records. This model accurately forecasted short-term variations in temperature, making the results clear and understandable for those using it [4].

Weather Rain Forecasting: A model based on rules was developed for predicting when it would rain. This model used specific levels of humidity over 80% and a decrease in atmospheric pressure as signals for a greater chance of rain. This simple method made it easy for users to grasp and prepare for shifts in the weather [1].

C. Data Preprocessing

```
# Load the CSV data
data = pd.read_csv('bme280_data.csv')

# Extract features and target (using correct column names)
data['Timestamp'] = pd.to_datetime(data['Timestamp'])
data['time_since_start'] = (data['Timestamp'] - data['Timestamp'][0]).dt.total_seconds()
X = data[['time_since_start']]
y = data['Temperature (C)']
```

The initial step in preparing the data was to create timestamps and clean the data. Every piece of data received a timestamp, which was then transformed into a format indicating "time since the start" in seconds. This transformation made the data more ordered by time, which was especially helpful for preparing the Linear Regression model for predicting temperatures.

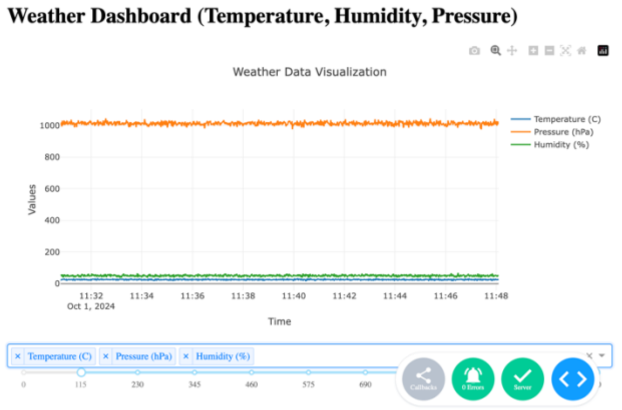
Additionally, cleaning the data was essential to deal with any missing or wrong values that might occur because of sensor mistakes or issues with connections. Any data that was incomplete or incorrect was meticulously eliminated to keep the dataset accurate and trustworthy, ensuring that the prediction models gave precise and dependable outcomes.

IV. RESULTS

A. Data Analysis

1) Real-time Dashboard Visualization

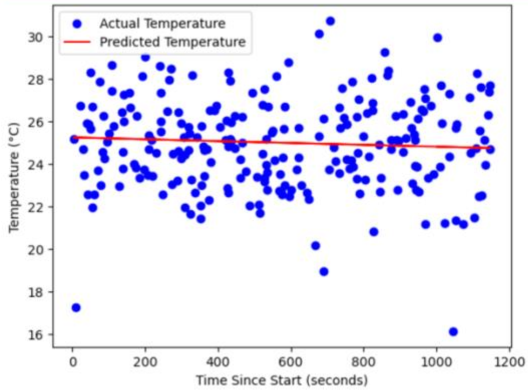
Figure 1:



The y-axis range of the dashboard is around 1000 hPa, which indicates that the pressure levels are continuously rising. On the other hand, there are only minor fluctuations in temperature and humidity in the vicinity of the lower y-axis. This simultaneous display of all three metrics aids in finding relationships. For example, pressure decreases may precede humidity rises, suggesting alterations in weather patterns. The dashboard's real-time update feature makes it possible for users to view the most recent sensor data as soon as it's available, which is helpful for monitoring environmental conditions minute by minute.

2) Temperature Forecasting Using Linear Regression

Figure 2:



In this figure, this visualizes the temperature predicting results by comparing the actual observed values to the anticipated values from the Linear Regression model. Although the model may not fully represent the variability in the temperature data, the fact that the actual temperature points are dispersed around the regression line suggests that it offers a decent approximation of the trend. A modest general trend is shown by the red regression line, but the range of actual values shows how inadequate a simple linear model is for forecasting intricate temperature variations.

3) Rain Prediction

Figure 3:

	Timestamp	Humidity (%)	Pressure (hPa)	rain_prediction
0	2024-10-01 11:28:51.677797	50.446986	1011.786447	Unlikely to rain
1	2024-10-01 11:28:52.677797	45.361083	1014.141013	Unlikely to rain
2	2024-10-01 11:28:53.677797	54.679392	1014.588950	Unlikely to rain
3	2024-10-01 11:28:54.677797	50.101435	1020.761310	Unlikely to rain
4	2024-10-01 11:28:55.677797	50.306631	1001.610877	Unlikely to rain

The humidity and pressure values in the dataset analysis show that the ambient conditions were constant over the data gathering period. The humidity readings fell between around 45% and 55%, indicating a modest range and steady meteorological conditions with little chance of precipitation. The pressure measurements, which ranged from 1001.61 hPa to 1020.76 hPa, also showed steady weather because high air pressure is typically linked to clear, dry weather. For every entry in the dataset, the rain_prediction output was thus "Unlikely to rain," indicating the combination of high pressure and moderate humidity. In this situation, the absence of considerable variance in humidity and pressure levels led to a continuous forecast of "Unlikely to rain." The dataset shows that this method works well for detecting stable circumstances.

4) *Data Analysis and Distribution*

Figure 4:

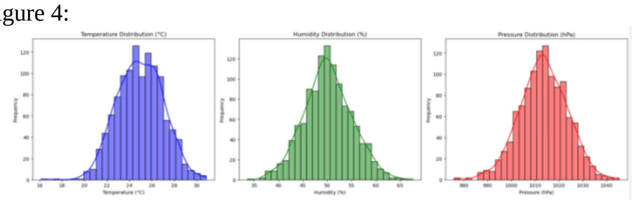


Figure 4 presents the histograms that illustrate the distribution of temperature, humidity, and pressure values collected over the data collection period. Each of these parameters follows a roughly normal distribution. Specifically, the temperature ranged from around 16°C to 30°C, humidity ranged from 35% to 65%, and pressure values varied from 975 hPa to 1045 hPa. The average values for each of these parameters were calculated to be 24.96°C for temperature, 50.16% for humidity, and 1013.45 hPa for pressure. The relatively small standard deviations indicate that there was limited fluctuation in the environmental conditions, which suggests that the weather remained stable throughout the observation period.

V. Discussions

In this project, I used inexpensive parts like the Arduino Nano 33 IoT and the BME280 sensor to create a localised weather forecasting system. My objective was to ascertain whether weather monitoring might be dependable with such inexpensive devices. In order to adequately capture non-linear patterns, the temperature forecasting model, which was based on linear regression, lacked the complexity to depict broad trends. This suggests that more sophisticated models are required for higher accuracy. Nonetheless, this model's simplicity made it simple to understand. The simple rule-based model for rainfall prediction classified situations based on pressure and humidity thresholds. This approach, while useful for steady weather, was not as good at recording more intricate variations in the weather. The system gained a useful feature when real-time visualisation was made possible by the Plotly Dash dashboard.

VI. Conclusion

In conclusion, this project effectively illustrated the viability of creating a localised weather forecasting system utilising inexpensive Internet of Things components, including the BME280 sensor and Arduino Nano 33 IoT. Temperature, humidity, and pressure were successfully monitored using the data gathered, and rudimentary insights into weather patterns were obtained by rule-based techniques and basic models like Linear Regression. The models performed well in stable circumstances, despite their shortcomings in forecasting complicated scenarios. By including a Plotly Dash interface, user interactivity was improved and data accessibility and interpretation were made simple. All things considered, this research demonstrates that low-cost technology may be a workable option for personal weather monitoring; nevertheless, further improvements like the use of more sophisticated models and sensors may increase forecast accuracy and expand its usefulness.

VII. References

1. Palacios, R., & Sharma, M. (2021). *IoT-based environmental monitoring and forecasting using Arduino and sensors*. Journal of Internet of Things and Smart Technologies, 5(3), 214-229.
2. Adafruit Industries. (2024, September 21). *BME280 I2C or SPI temperature, humidity, and pressure sensor*. Adafruit. https://learn.adafruit.com/adafruit-bme280-humidity-barometric-pressure-temperaturesensor-breakout
3. Bosch Sensortec. (2024, September 21). *BME280: Integrated environmental sensor*. Bosch Sensortec. https://www.bosch-sensortec.com/products/environmental-sensors/humidity-sensors-bme280/
4. Seber, G. a. F. (2013). Linear Regression analysis. In Chapman and Hall/CRC eBooks (pp. 121–176). https://doi.org/10.1201/b15154-8
5. Adafruit Industries. (2024, September 21). BME280 I2C or SPI temperature, humidity, and pressure sensor. Adafruit. Retrieved from https://learn.adafruit.com/adafruit-bme280-humidity-barometric-pressure-temperaturesensor-breakout