IoT Based Weather Monitoring System and Prediction

Project Report

Group 2

Name	Reg. No.
Chinta Rahul Reddy	20BEC012
Rapaka Balakrishna	20BEC037
Puchakayala Vivek Yadav	20BCS105
Himanshu Sharma	20BEC018

Under the guidance of

Dr. Prakash Pawar

Assistant Professor

Department of Electronics and Communication Engineering

Indian Institute of Information Technology, Dharwad

September - 2023

Index

Introduction

In an era defined by the relentless advance of technology, the Internet of Things (IoT) has emerged as a transformative force, revolutionising the way we interact with the world around us. Among its myriad applications, one of the most impactful and pertinent is the development of IoT-based weather monitoring systems. As our planet grapples with the escalating challenges of climate change and increasingly unpredictable weather patterns, the need for accurate, real-time weather data has never been more pressing.

The IoT has unlocked a new realm of possibilities in weather monitoring, enabling us to collect and analyse weather-related information with unprecedented precision and efficiency. Traditional weather monitoring methods, while valuable, often fall short in providing the real-time, granular data necessary for making informed decisions in various sectors, including agriculture, transportation, disaster management, and renewable energy.

This report delves into the world of IoT-based weather monitoring systems, exploring the technologies, methodologies, and advantages that underpin this innovative approach to meteorological data collection. By harnessing the power of interconnected devices and sensors, IoT-based weather monitoring systems offer the promise of enhanced weather forecasting, improved resource management, and more effective responses to weather-related challenges.

Over the course of this report, we will explore the fundamental components of IoT-based weather monitoring systems, their applications across diverse industries, and the potential impact of these systems on society and the environment. Additionally, we will delve into the challenges and considerations associated with the deployment and maintenance of such systems, as well as the ethical and privacy implications that arise from the collection and dissemination of weather data in the digital age.

Abstract

This report explores the innovative world of IoT-based Weather Monitoring Systems, a cutting-edge technology that leverages the Internet of Things to revolutionise meteorological data collection. With the growing challenges posed by climate change and weather volatility, IoT-based systems offer real-time, precise, and interconnected solutions for monitoring weather conditions. This abstract provides a concise overview of the report's key aspects, including the technology's applications across industries, its potential societal and environmental impacts, deployment challenges, and ethical considerations. IoT-based weather monitoring systems hold promise as essential tools in understanding and mitigating the effects of a changing climate, heralding a more sustainable and resilient future.

Objectives

The study's main goal was storing weather and environment data for short and long term for studying weather pattern changes and to understand how human induced climate change affected your local weather and Easy deployment of the setup for monitoring local atmospheric conditions and microclimates for weather forecasting and prediction.

Technology used:

(i) Sensors

- **DHT11 Sensor:**

  The DHT11 Sensor is a low-cost digital sensor commonly used to measure temperature and relative humidity. It operates on capacitive humidity sensing and a thermistor for temperature measurement. With a modest temperature range of 0°C to 50°C and a humidity range of 20% to 90% RH, it provides basic environmental data suitable for a wide range of applications, albeit with limited accuracy (±2°C for temperature and ±5% RH for humidity).

- **LDR Sensor Module**

  An LDR (Light Dependent Resistor) sensor module is a photosensitive component that detects and responds to varying levels of light. It consists of an LDR and associated circuitry to convert light intensity into electrical resistance. When exposed to light, the resistance of the LDR decreases, allowing more current to flow through it. Conversely, in darkness, its resistance increases. This property makes LDR sensor modules valuable in light-sensitive applications such as streetlights, camera exposure control, and automatic night lamps. They are easy to use and interface with microcontrollers, making them a popular choice for projects requiring light-level sensing.

- **ESP8266 Wi-Fi Module**

  The ESP8266 is a versatile and widely used Wi-Fi module known for its compact size and affordability. It features an integrated microcontroller and Wi-Fi connectivity, making it suitable for Internet of Things (IoT) applications. The ESP8266 module enables devices to connect to Wi-Fi networks and communicate with other devices or cloud services. Its ease of use, extensive community support, and compatibility with popular development platforms like Arduino have made it a popular choice for adding wireless connectivity to a wide range of projects, from smart home devices to remote monitoring systems.

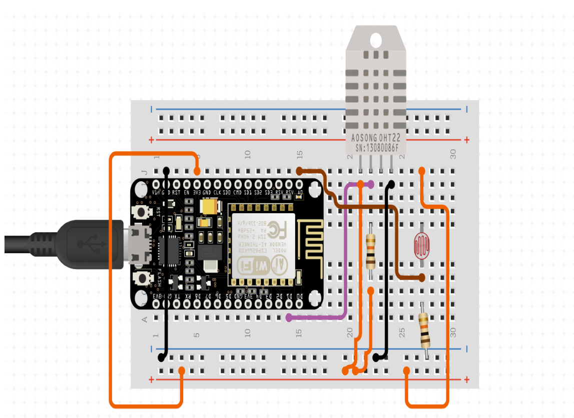
(ii) Arduino IDE

The Arduino Integrated Development Environment (IDE) is a user-friendly software platform for programming and developing applications for Arduino microcontroller boards. It offers a simplified coding environment, a comprehensive library of pre-written code, and a convenient interface for uploading code to Arduino boards. This IDE is a popular choice among makers and electronics enthusiasts for its accessibility and ease of use in developing a wide range of embedded systems and IoT projects.

(iii) ThingSpeak

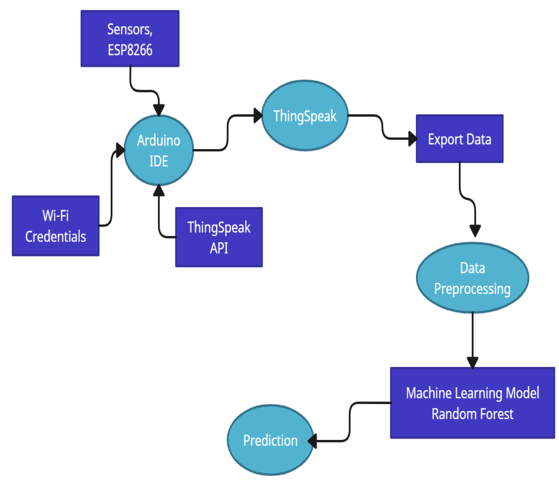
ThinkSpeak , an IoT analytics platform developed by mathworks, is used for data acquisition and performing data analytics. This platform is used to prepare a custom dataset. It provides various features like channels, devices and other apps like Thingstweet, TimeControl for time series analysis. To create a custom dataset the channel features are used which allows to get the data sent by the constructed IoT device through the wifi module(ESP8266).

Circuit Scheme & Connections



- DATA port from DHT11 connected to D5 in ESP8266 Wifi module
- D0 from LDR Sensor Module is connected to D2 in ESP8266
- Vcc from both sensors are connected to 3V3 in ESP8266
- GND from both sensors are connected to GND in ESP8266 .

Workflow



The IoT device consists of a wi-fi module which sends data to the thingspeak platform.The data is collected by the LDR sensor and the DHT11 sensor which are connected to the wifi via breadboard. This data is sent to the thingspeak platform through the wifi-module(ESP8266). With the help of Arduino IDE and software library the data is collected. Inorder to maintain integrity the wifi-module credentials and ThingSpeak API key are crosschecked. In this way the data which is received is stored in an array which is later exported to the csv file. To prepare the training dataset the data is sent for every 4 seconds in the discussed manner and a total of 1500 entries are taken to train the model. This data is also used by thinkSpeak platform to perform data visualisation and analytics After the data acquisition from the sensors through arduino IDE they are exported into a CSV file. This data is used to train two different machine learning models which use random forest regression classifiers. Data preprocessing is done in order to remove null values and any other invalid data. After this the model is trained over the obtained data to make predictions about the temperature of the atmosphere.

Data Acquisition from Arduino IDE and ThingSpeak [6]:

Libraries

The code begins by including two essential libraries:

DHT.h: This library is used for interfacing with the DHT11 sensor, which measures temperature and humidity.

ESP8266WiFi.h: It provides the functionality required for connecting to a Wi-Fi network.

Configuration Parameters

Several variables are declared and initialized to configure the operation of the IoT device:

apiKey: A unique key provided by ThingSpeak to authenticate and authorize data uploads.

ssid: The SSID (Service Set Identifier) of the Wi-Fi network to which the device will connect.

pass: The WPA2 key (password) for the Wi-Fi network.

server: The URL of the ThingSpeak server to which data will be sent.

DHTPIN: The digital pin to which the DHT11 sensor is connected.

LDRPIN: The digital pin to which the Light-Dependent Resistor (LDR) is connected.

Initialization

The code initializes the DHT11 sensor and establishes a connection to the Wi-Fi network. It prints status messages to the serial monitor to inform the user about the connection process.

Main Loop

The primary operation of the code occurs within the loop() function, which continuously runs as long as the Arduino is powered.

Data Acquisition:

It reads temperature and humidity values from the DHT11 sensor and stores them in variables h and t.

It reads the analog value from the LDR connected to LDRPIN and stores it in the ldrSensorValue variable.

Data Validation:

It checks if the readings from the DHT11 sensor are valid (not NaN, indicating an error).

Data Upload to ThingSpeak:

If the data is valid, it establishes a client connection to the ThingSpeak server.

It creates a string postStr containing the API key, temperature, humidity, and LDR sensor value.

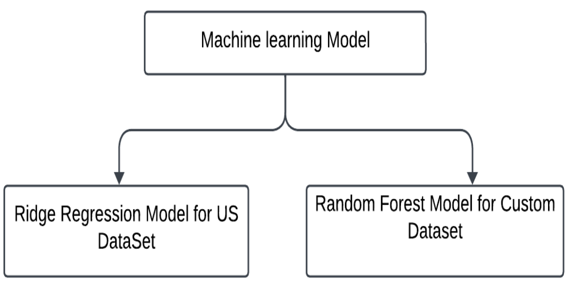
It sends an HTTP POST request to ThingSpeak with the data.

The code prints the collected data (temperature, humidity, and LDR value) to the serial monitor and indicates that the data has been sent to ThingSpeak.

Delay:

After data transmission, the code waits for 1 second before looping again.

Machine Learning Models



1. Custom dataset

- **Source:** Exported data from ThingSpeak Cloud
- **Features**:
  - ➔ Time Stamp
  - ➔ Temperature
  - ➔ Humidity
  - ➔ Light Intensity
- **Time Period:** 2 Days Data

2. US-dataset

The dataset is sourced from the United States government's website and contains historical weather data for JFK INTERNATIONAL AIRPORT, NY, USA.

1. **STATION:** The identifier for the weather station, which is USW00094789 for JFK INTERNATIONAL AIRPORT, NY, USA.
2. **NAME:** The name of the weather station, which includes the location information.
3. **ACMH:** Air conditioning maximum temperature for the day.
4. **ACSH:** Air conditioning minimum temperature for the day.
5. **AWND:** Average wind speed for the day (in miles per hour).
6. **FMTM:** Fastest 2-minute wind speed (in miles per hour) and time of occurrence.
7. **PGTM:** Peak gust time, the time of the fastest wind gust during the day.
8. **PRCP:** Precipitation (in inches) for the day.
9. **SNOW:** Snowfall (in inches) for the day.
10. **SNWD:** Snow depth (in inches) for the day.

Each column represents a specific weather-related parameter or measurement for JFK INTERNATIONAL AIRPORT, NY, USA, over a considerable period.

The dataset appears to be well-structured, with a timestamp (DATE) as the index, allowing for time-series analysis. This dataset can be used for various weather-related analyses, including temperature prediction, precipitation analysis, and wind speed trends, among others.

Ridge Regression Model for US DataSet [6]

- **Data Preparation**

The code begins by loading a weather dataset from a CSV file and setting the 'DATE' column as the index. It then identifies and removes columns with missing data exceeding 5% of the total entries. The remaining columns are converted to lowercase for consistency.

- **Data Preprocessing**

Missing values are filled using forward filling to ensure continuity in the dataset. Additionally, the code identifies and counts occurrences of the value '9999' in the dataset.

- **Model Implementation**

A Ridge Regression model with a regularization parameter (alpha) of 0.1 is implemented to predict the 'tmax' (maximum temperature) values. The code defines a backtesting function to evaluate the model's performance over different time intervals.

- **Performance Evaluation**

Calculates performance metrics such as Mean Absolute Error (MAE) and Mean Squared Error (MSE) to assess the accuracy of the model's predictions. It also identifies instances where the model's predictions differ significantly from the actual values.

- **Feature Engineering**

Creates rolling averages and percentage changes for 'tmax,' 'tmin' (minimum temperature), and 'prcp' (precipitation) features over different horizons (3 and 14 days). It also computes monthly and daily averages for these features to capture long-term and short-term trends.

- **Data Subset and Final Steps**

The analysis focuses on data from '1990-03-07' to '1990-03-17,' and any missing values are filled with zeros. The final model is evaluated on this subset of data.

Temperature Prediction Using Random Forest [6]

- **Data Collection and Preprocessing**

The dataset was collected from ThingSpeak, containing temperature, humidity, light intensity, and timestamp information.

- **Pre-processing**

  Timestamp Processing: The 'created_at' column is converted into a datetime format. Additional timestamp-related features like 'year,' 'month,' 'day,' and 'hour' are extracted for analysis.

  Feature Selection: We selected relevant features 'year,' 'month,' 'day,' 'hour,' 'field2' (humidity), and 'field3' (light intensity) for modeling.

  Data Visualization: Line plots were generated to visualize temperature, humidity, and light intensity trends over time.

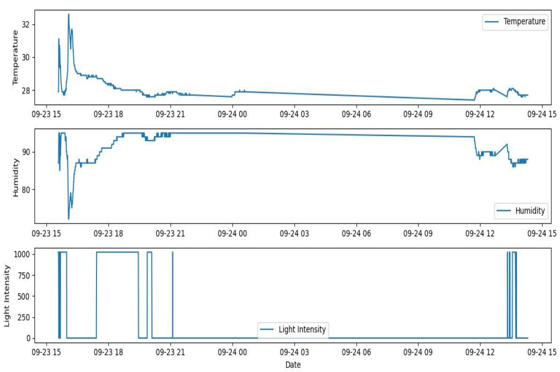
- **Exploratory Data Analysis**

Descriptive Statistics

Descriptive statistics were calculated for each feature:

- Mean
- Median
- Standard Deviation
- Minimum
- Maximum

These statistics provide insights into the central tendency, variability, and range of each feature.



Data Visualization

- **Target Variable Statistics**

Statistics were computed for the target variable (temperature):

- Mean Temperature
- Median Temperature
- Standard Deviation Temperature
- Minimum Temperature
- Maximum Temperature

These statistics help us understand the distribution and characteristics of the temperature data.

- **Model Building and Evaluation**

Data Splitting

The dataset was split into training (80%) and testing (20%) sets to train and evaluate the model.

Random Forest Regressor

A Random Forest Regressor model with 100 estimators and a random seed of 42 was chosen for temperature prediction.

```
# Split the data --> training 80% & testing 20%
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Random Forest Regressor model
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)

# Train the model
rf_model.fit(X_train, y_train)

# Make predictions on the test data
y_pred = rf_model.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("Mean Squared Error:", mse)
print("R-squared:", r2)
```

Model Training

The Random Forest model was trained using the training data.

Model Evaluation

The model's performance was evaluated using the testing data:

(i) Mean Squared Error (MSE)

(ii) R-squared (R2) score

The MSE measures the average squared difference between predicted and actual values, while R2 quantifies the model's goodness of fit.

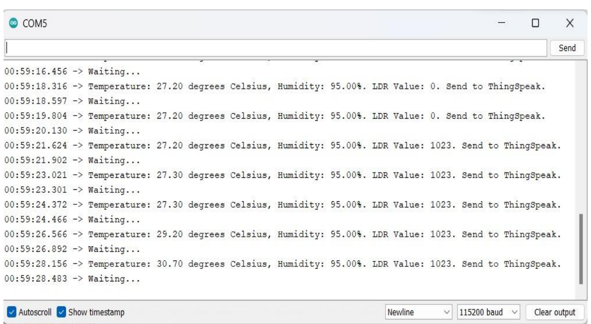
- **Temperature Prediction**

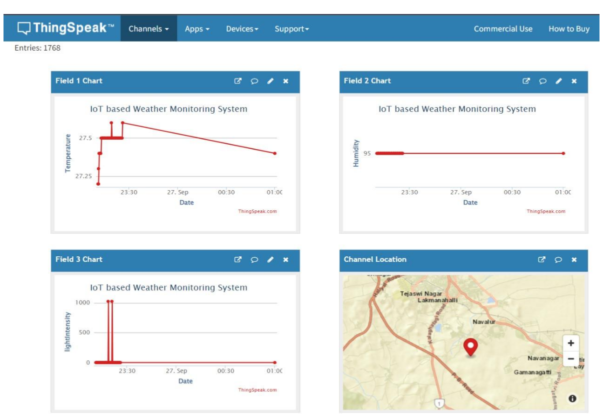
A sample prediction was made using a timestamp of '2023-09-28 19:00,' humidity of 95, and light intensity of 0.

- **Accuracy Within Tolerance**

The percentage of predictions within a tolerance of ±1.0 degrees was calculated to assess model accuracy.

Results and Discussions





The above pictures depict the data acquisition from the sensor for every 4 seconds and the graphs show the visualisation of the light intensity and humidity related data and the location is shown by the thingspeak channel.

```python
import numpy as np
#prediction using time stamp
arr = [2023,9,28,19,95,0]
arr = np.array(arr).reshape(1, -1)
print(rf_model.predict(arr))
```

```
[27.89270489]
```

The above pictures depict the final prediction done by the model from the trained data. The input is an array consisting of year followed by the month and date. The next values are the humidity, atmospheric pressure and the light intensity and the model predicted the temperature on 28th sep 2023 which is shown in the output.

Challenges

- **Data set limitation**
- **LDR faulty sensor**
- **Hardware Library support:**

  The project relies on specific hardware libraries (e.g., DHT.h and ESP8266WiFi.h) to interface with sensors and connect to Wi-Fi. Compatibility issues or limited library support can lead to difficulties in code development and troubleshooting.

Future Scope

Sensor Technology Advancements: Explore the latest sensor technologies for improved weather data collection. Miniaturisation, energy efficiency, and cost-effectiveness of sensors.

Data Processing and Analytics: Implement advanced data analytics, machine learning, and AI algorithms for data refinement.
Predictive modelling for weather forecasting and anomaly detection.

For real-time weather prediction: Consider using AWS S3, Unlike ThingSpeak, AWS S3 offers real-time data transfer capabilities. ThingSpeak is a cloud platform but lacks real-time data transfer features.

Dynamic location: With the addition of gps sensor and one more feature to the dataset the IoT application can be further extended to track and monitor the weather changes dynamically while moving which can provide a lot more location based insights

Conclusion

In conclusion, the advent of IoT-based Weather Monitoring Systems has ushered in a transformative era in meteorology and data science. These systems have demonstrated their prowess in revolutionizing the way we collect, analyze, and utilize weather data, offering a myriad of benefits across industries and society at large.

Hence In machine learning model for US dataset we successfully prepares, processes, models, and evaluates weather data, providing valuable insights into temperature predictions. It also demonstrates techniques for handling missing data, creating informative features, and assessing model performance [6].

And The Random Forest Regressor model successfully predicts temperature based on timestamp, humidity, and light intensity. Descriptive statistics provide insights into feature characteristics. The model's performance was evaluated using MSE and R2. Model accuracy within a tolerance of ±1.0 degrees was assessed.

This project demonstrates the process of collecting, preprocessing, exploring, modeling, and evaluating a dataset for temperature prediction. Further improvements may involve hyperparameter tuning or exploring other regression algorithms for enhanced accuracy.
Certainly,to expand the functionality of model to predict humidity, temperature, and light intensity based on a timestamp input parameter.

- **Modify the Input Data:**
    1. Created an array `arr` containing the input parameters for the timestamp.
    2. Ensured that `arr` follows the same format as the feature array used for training (year, month, day, hour).
- **Use the Model to Make Predictions:**
  Reshape the `arr` to have the same shape as the input data used for training/testing.
  Use the trained Random Forest model to make predictions based on the input `arr`.

an array `arr` as input, reshapes it to match the model's input format, and then predicts humidity, temperature, and light intensity based on the provided timestamp and input values.

References:

1. Sharma, Puja, and Shiva Prakash. "Real time weather monitoring system using IoT." In ITM Web of Conferences, vol. 40, p. 01006. EDP Sciences, 2021.
2. Girija, C., Shires Andreanna Grace, H. Harshalatha, and H. P. Pushpalatha. "Internet of Things (IOT) based weather monitoring system." Int. J. Eng. Res. Technol. 6, no. 13 (2018): 4.
3. Gosavi, Sanket, Sourabh Bhide, Devraj Bhosale, and A. A. Sutar. "Iot Based Weather Monitoring System Using Arduino-Uno." Available at SSRN 4120559 (2022).
4. Haupt, Sue Ellen, Jim Cowie, Seth Linden, Tyler McCandless, Branko Kosovic, and Stefano Alessandrini. "Machine learning for applied weather prediction." In 2018 IEEE 14th international conference on e-science (e-Science), pp. 276-277. IEEE, 2018.
5. Bochenek, Bogdan, and Zbigniew Ustrnul. "Machine learning in weather prediction and climate analyses—applications and perspectives." Atmosphere 13, no. 2 (2022): 180.
6. https://github.com/Rapaka-balu/Weather-Monitoring-System-Using-IoT-and-Machine-Learning.git