International Journal on Science and Technology (IJSAT)

E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

IOT-based Rain Fall prediction

Raghuveer Kumar[1], Anuj Poonia[2], Ravinder[3], Gaurav Soni [4]

[1,2,3]AIT-CSE, Chandigarh University, Mohali, Punjab, India
[4]Associate Prof., Department of AIT-CSE Chandigarh University Mohali, Punjab, India

Abstract

In this project we made an IoT-based rainfall prediction system. Which provides you a realtime monitoring and forecasting system of rainfall using different types of sensors and different microcontroller technology. In this system we use a DHT11 sensor for compute the temperature and humidity of the environment, a BMP280 sensor is used for reading the pressure and rain sensor for rain detection and precipitation.All the sensor is connected to the Arduino Uno and the Breadboard. Which processes the whole environmental data and gives the reading. This is a low-cost energy–efficient project which is very useful for rural area issues like agriculture and it help the urban society to plan their disaster management.This project helps the rural and urban areas to take smart decision on the basis of this project.

1. Introduction

Rainfall prediction is a main aspect of meteorology, with straight through implications for agriculture, water management, and disaster preparation. Accurate forecasting can help minimize losses due to unexpected weather events, optimize irrigation planning, and support sustainable development practices. However, traditional weather forecasting models often depend on large-scale meteorological stations and satellite data, which may not always offer high-resolution, real-time information at the local level.

Over the last few years, the IoT has provided new opportunities for the environmental monitoring by using the sensor-based systems. Smart weather monitoring station using IoT is more effective, efficient, cheap, and can easily be expanded. As these sensors are compact and inexpensive, the environmental data can be collected with high temporal resolution, allowing for more localised and timely predictions to be made.

This project deals with the creation of IoT based rainfall prediction system. Temperature and humidity are measured by the DHT11 sensor, while the BMP280 sensor measures the atmospheric pressure, and the rain sensor determines the start of the rainfall. These are connected to an Arduino Uno microcontroller that acts as the data acquisition and processing module. The gathered information is then processed in order to determine the probability of rain and can be outputted on a connected computing device or stored in a cloud for access from another computing device.

The proposed system shall use real-time data acquisition with basic data analytics to generate timely rainfall alerts particularly in areas with little or no access to better forecasting infrastructure. This system is ideal in precision farming and in disaster risk reduction at the community level.

2. Literature Review

Rainfall prediction has been a topic of interest for a long time because of its importance in agriculture, hydrology, urban planning and disaster management. Rainfall prediction has been done previously using large-scale meteorological data, satellite data, and other numerical weather prediction (NWP) models. Although these approaches have been useful at the regional and global scales, they are not sufficient to provide the detailed, timely, and location-specific information required for precision agriculture or community-level decision making. The development of the IoT in the last few years has made it possible to have more distributed and sensor-based systems for weather monitoring that can improve the accuracy of rainfall prediction.

2.1 Traditional Approaches to Rainfall Forecasting

Current operational models like GFS and ECMWF rely on numerous parameters and algorithms obtained from satellites, radar, and ground stations. These models usually employ mathematical equations to describe the behaviour of the atmosphere with the aid of parameters such as humidity, temperature, wind and pressure (Kalnay et al., 1996). Despite the fact that these models provide useful information for regional prediction, the resolution and the update rate of these models are not sufficient for the micro- level prediction. However, the infrastructure needed for such models is costly and unavailable to the resource- scarce regions.

To overcome this, some works have tried to use past climate information for the training of statistical models. Some of these models include regression analysis, Markov chain models, and autoregressive integrated moving average (ARIMA) models have been used in short-term rainfall prediction (Box & Jenkins, 1976). Nevertheless, these models are less effective in the analysis of climate data and are not very accurate in the case of sudden changes in the climate.

2.2 Emergence of IoT in Environmental Monitoring

The drawbacks of traditional forecasting systems have forced researchers to turn to IoT solutions, which include sensors, microcontrollers, and data transmission modules in order to develop distributed and real-time monitoring systems. The benefits of IoT include the ability to be affordable, portable, easy to deploy and scalable. These systems are particularly helpful in the rural or the remote areas where there is no physical structures.

Kumar and Patel (2018) proposed an IoT based weather station with sensors for temperature, humidity, and rainfall and use cloud server for data visualisation. Their system also proved that it is possible to construct low cost monitoring tools using available hardware resources. In the same year, Rajalakshmi and Anuradha suggested the development of a smart irrigation system that uses soil moisture sensors and weather data from an IoT station to optimise irrigation times and minimise water wastage for better crop yields.

Patil and Deshmukh (2020) have also described how Arduino-based weather systems are implemented in the following ways. They developed a prototype by integrating DHT11 and BMP180 sensors for measuring environmental conditions and a rain sensor. They established that such systems could benefited local farmers through timely warning and forecast.

2.3 Sensor Technologies for Rainfall Prediction

The key to any prediction system based on IoT is the selection and incorporation of sensors. Temperature and humidity are basic parameters as the second indicates that high humidity level is likely to be followed by rain. The DHT11 sensors are used due to their low cost and easy to use. Another important parameter is the atmospheric pressure, which can be measured with the help of BMP280 or BMP180 sensor; the sharp decrease in pressure means that it will soon start raining.

For instance, there are the YL-83 or FC-37 which are mainly used for direct measurement of precipitation. However, they are useful in determining the occurrence of rainfall but they cannot predict the occurrence of rainfall without the help of other environmental factors. Sharma and Bansal (2020) pointed out that the integration of multiple sensors such as pressure and humidity enhances the detection of rain trend in the early hours.

Real-time data collection, analysis, and transfer are possible through the connexion of these sensors with microcontrollers such as Arduino Uno or ESP8266. In the recent past, the development of IoT based monitoring systems has been made very easy by the use of open source platforms and libraries for interfacing the sensors.

2.4 Machine Learning in Rainfall Forecasting

While basic sensor-based systems provide only an alert, an application of ML can provide predictive information. Some of the related researches have investigated the use of ML algorithms such as the support vector machines (SVM), decision trees, and artificial neural networks (ANN) in forecasting rainfall from the recorded sensor data.

Gupta et al., (2021) developed an ANN model based on sensor data of an IoT weather station and enhanced the accuracy of the rainfall prediction. That is their system showed that the integration of real-time IoT data with analytical models could perform better than threshold- based systems. However, there is still the problem of getting enough training data, especially in the rural areas where there are few such data history.

Classification algorithms and their performance were discussed in another study by Meena and Choudhary (2022) on weather datasets. They also deduced the point that the effectiveness of the ML models is based on the data pre-processing, the selection of the features and the amount of historical data used in the model.

However, the application of ML-based systems is not without its challenges as they are usually resource-intensive and demand internet connexion and storage which may not be easily accessible in some environments. This is a limitation especially when there is a need to have quick and timely alerts in areas where infrastructure is not well developed.

2.5 Comparative Analysis and Gaps in Literature

Based on the review of the current literature, it can be noted that there are many IoT-based weather systems proposed in the literature, however, little effort has been devoted to a rainfall prediction system that requires minimal hardware and data at a localised level. All the current systems are just simple weather stations that gather data but do not help in decision making and do not issue alerts.

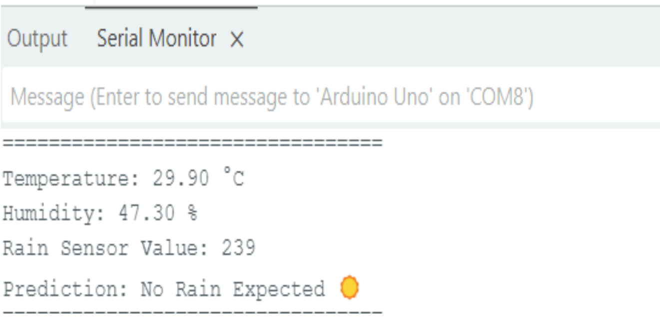
However, there are many researches that are concerned with data gathering and mapping but do not use multiple sensor parameters to reason. There is also little consideration given to the durability and reliability of the products when used under different weather conditions. For instance, the system developed by Das and Roy (2020) was tested for only a few days which was not enough to ascertain the performance and reliability of the system.

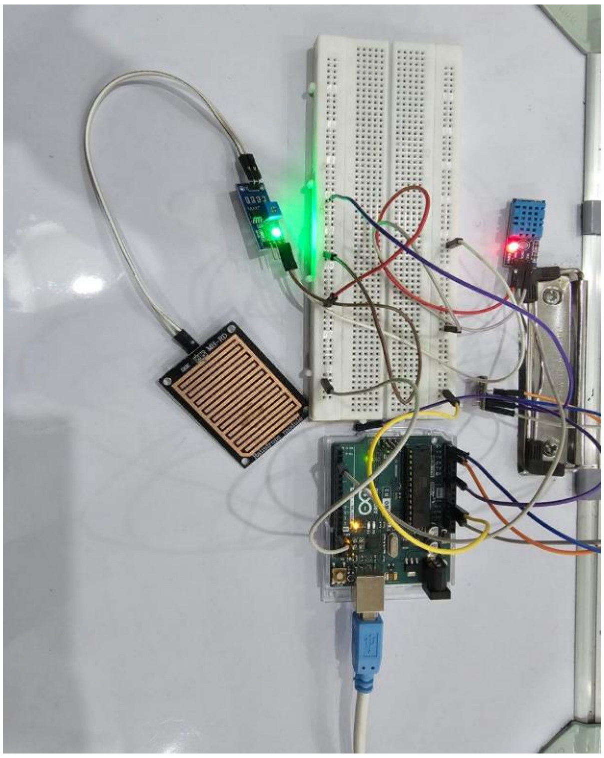
One of the other gaps is the absence of the modular and easily scalable models that can be implemented at the community level without the need for professional IT solutions. On the one hand, cloud-based platforms bring new functionalities that make the work of an organisation more efficient, on the other hand, they bring the problem of dependence on the Internet connexion and electricity, which in many cases is unstable in rural areas.

2.6 Relevance of the Proposed Work

Based on these observations, the proposed research seeks to design a cheap, easy-to-implement and effective rainfall prediction method using few IoT devices. This involves using temperature, humidity, pressure and rain sensors to collect real time data and set logical thresholds in order to predict the probability of rainfall. Unlike other data-based models, the system uses light logic, which is compatible with Arduino Uno microcontrollers and their equivalents.

In its applicability, it is applied mainly in rural and farming sectors, and it is especially designed to be simple, dependable, and inexpensive. This way it does not only covers the gap between mere weather observing and complex forecasting systems but also provides affordable tools to help communities to be more prepared for weather- related events.





Methodology

The IoT rainfall prediction system was developed through several phases, starting from choosing the components to acquire real-time data. The aim was to design a miniature, inexpensive and effective structure to monitor environmental factors related to rainfall and send the collected data to be analysed.

4.1 Component Selection

The core of the system consists of four key components:

- **Arduino Uno**: Serves as the main microcontroller for data acquisition and device control.
- **DHT11 Sensor**: Measures ambient temperature and humidity levels, both of which are critical indicators of atmospheric moisture.
- **BMP280 Sensor**: Records barometric pressure and temperature, which help in predicting weather changes.
- **Rain Sensor Module (YL-83)**: Detects the presence and intensity of rainfall based on surface conductivity.

These components were selected for their affordability, reliability, and ease of integration with microcontrollers.

4.2 Circuit Assembly

All sensors were connected to the breadboard and connected with Arduino Uno through the jumper wires. The DHT11 and BMP280 were connected to digital pins and I2C interface while the rain sensor was connected to an analogue pin. The whole system was connected through a USB cable from a laptop that also acted as the data receiver and visualizer.

4.3 Data Acquisition

Each sensor was programmed to collect data at intervals of 10 seconds. The Arduino was loaded with a custom script written in C++ using the Arduino IDE. Sensor readings were continuously fetched and temporarily stored for immediate display. In particular:

- The DHT11 returned temperature and humidity in real time.
- The BMP280 captured atmospheric pressure variations.
- The rain sensor output ranged from 0 (no rain) to 1023 (heavy rain), based on moisture level.

4.4 Data Transmission and Display

Sensor data was transmitted serially to the laptop, where it was displayed using the Serial Monitor tool. For further development, the system could be integrated with cloud platforms like ThingSpeak or Blynk for remote monitoring and graphical visualization.

4.5 Prediction Logic

The rainfall prediction was based on thresholds derived from meteorological observations:

- A rapid drop in barometric pressure coupled with rising humidity signaled an increasing chance of rainfall.
- Detection by the rain sensor confirmed precipitation, which was then logged as rainfall.
- The system compared current readings with preset conditions to output a simple forecast message

—“No Rain,” “Light Rain Likely,” or “Heavy Rain Possible.”

4.6 Testing and Calibration

Multiple tests were carried out in varying weather conditions to verify sensor accuracy. Minor calibration adjustments were made using offset values in the Arduino code. Rain sensor sensitivity was also tweaked to prevent false triggers due to dew or surface moisture.

3.2 Hardware Components

The key hardware components used in the system are listed below:

- Arduino Uno: A microcontroller board based on the ATmega328P, selected for its simplicity, wide community support, and sufficient I/O capabilities for interfacing multiple sensors.
- DHT11 Sensor: A digital sensor used to measure temperature and relative humidity. It provides data in a digital format, minimizing the need for analog signal conversion.
- BMP280 Sensor: A high-precision barometric pressure sensor capable of measuring atmospheric pressure and temperature. This sensor is essential for detecting pressure drops, which are often precursors to rainfall.
- Rain Sensor (YL-83): A resistive-type sensor that detects the presence of water droplets on its surface. It is primarily used to confirm precipitation events and support rainfall detection.
- Laptop/Cloud Platform (Optional): For data visualization, logging, or remote monitoring, the system can be interfaced with a laptop via USB or with cloud services via Wi-Fi modules such as the ESP8266 (if added).
- Breadboard and Jumper Wires: Used for prototyping and establishing electrical connections among components.
- Power Supply: A USB connection or external 5V DC source powers the Arduino and connected sensors.

3.3 Sensor Integration and Interfacing

Each sensor is interfaced with the Arduino Uno using digital or I2C communication protocols:

- The DHT11 sensor uses a single digital pin for bi- directional communication. The Arduino reads humidity and temperature data every 10 seconds.
- The BMP280 sensor communicates via I2C, enabling the Arduino to retrieve atmospheric pressure and ambient temperature with high accuracy.
- The rain sensor is connected to an analog input pin. The sensor’s analog output voltage decreases in the presence of water, allowing the system to distinguish between dry and rainy conditions.

Sensor readings are obtained using dedicated Arduino libraries (DHT.h, Adafruit_BMP280.h) that facilitate data acquisition and calibration.

3.4 Data Acquisition Process

The data acquisition process begins with sensor initialization and a short warm-up period to stabilize readings. Once initialized, the system enters a continuous monitoring loop, sampling sensor data at regular intervals (e.g., every 10 seconds). The steps are as follows:

1. Read temperature and humidity from DHT11.
2. Read atmospheric pressure from BMP280.
3. Read analog value from the rain sensor.
4. Store all sensor values in memory.
5. Pass the data to the prediction module for evaluation. This loop ensures near-real-time data

monitoring with a latency of under 2 seconds per cycle.

3.5 Prediction Algorithm

The rainfall prediction logic is designed as a rule-based decision system, suitable for microcontroller-level implementation. Instead of using complex machine learning models, the system evaluates environmental conditions based on threshold values derived from empirical observations and prior research.

Decision Criteria:

- Humidity Threshold: ≥ 80%
- Atmospheric Pressure Threshold: ≤ 1000 hPa
- Rain Sensor Value Threshold: Analog reading < 400 (indicating presence of water)

Prediction Rules:

- If humidity ≥ 80% AND pressure ≤ 1000 hPa → High Probability of Rainfall
- If rain sensor is wet → Rainfall Detected (Confirmed)
- If only one of the above is true → Low Probability of Rainfall
- If none are true → No Rainfall Predicted

This logic ensures that predictions are made conservatively, reducing false positives while still offering timely alerts.

3.6 Output Generation and Display

The prediction outcome and sensor readings are displayed in real-time on a serial monitor via USB interface. The following information is printed:

- Temperature (°C)
- Humidity (%)
- Pressure (hPa)
- Rain sensor status (Dry/Wet)
- Prediction outcome (Rain Likely / No Rain / Rain Detected)

For extended use, the system can be expanded to log data on an SD card or upload it to cloud platforms like ThingSpeak, Blynk, or Firebase using additional communication modules (e.g., ESP8266 or GSM).

3.7 Data Logging and Analysis

To evaluate the performance of the system, sensor data and prediction outputs are logged over a fixed observation period (e.g., 14 days). This data is compared with actual weather conditions recorded manually or verified through online weather services to assess the system's accuracy.

Key metrics tracked include:

- True Positive Rate (correctly predicted rainfall)
- False Positive Rate (predicted rain but no actual rain)
- False Negative Rate (missed rainfall event)
- System response time (delay between sensor detection and prediction)

3.8 Calibration and Tuning

Sensor thresholds are subject to environmental variation, and initial calibration is essential. The following procedures are applied:

- Humidity and temperature values from DHT11 are cross-verified with local meteorological data for consistency.
- BMP280 readings are calibrated against standard pressure values at sea level and adjusted for altitude if needed.

- Rain sensor thresholds are determined experimentally by exposing the sensor to controlled water droplets and dry conditions.

Future iterations of the system may include dynamic threshold adjustment based on historical data or user feedback.

3.9 Limitations and Scope for Enhancement

While the system provides effective real-time monitoring, its predictive accuracy is constrained by its simple logic model. Additional parameters such as wind speed, cloud cover, or solar radiation could enhance reliability. Integrating a basic machine learning model is feasible for advanced deployments with better processing power and data availability.

For remote use, solar power and wireless communication modules can be incorporated to ensure autonomous and long-term field operation.

Conclusion & Future Scope

This paper demonstrates the successful implementation of an IoT-based system that is inexpensive and portable enough for predicting rainfall. In this case, the system incorporates environmental sensors such as temperature, humidity, atmosphere pressure, and rainfall sensors to provide a practical solution for monitoring the environment as well as short-term predictive weather conditions. By making the Arduino Uno the central controller, the implementation is made relatively easy, and the system can be easily used by the end users who may not have considerable expertise in programming.

From the results obtained in the experiment, it is clear that the logic-based prediction method that has been implemented in this setup is quite simple but effective in giving signals of the next possible rainfall event. It does this through the application of computed thresholds on the real-time obtained sensor data, thus it does not require large data sets or complex algorithms. This makes the system very ideal to be implemented in the rural areas or even areas that may not have any kind of weather infrastructure.

However, as it has been pointed out earlier, the current system is efficient in performing its basic tasks and functions. The prediction logic for example may be adaptive and learn from previous data to make it more accurate than before. This is the reason why incorporating more sensors like the wind speed or solar radiation could also be incorporated to enhance the decision making of the system. Further development can be made on the system to include a wireless communication interface to transfer the data received to other wireless devices or cloud based systems for monitoring. This would also help in its deployment in the off-grid areas because it operates through solar power. With such enhancements, this IoT based rainfall prediction model is likely to have a strong impact on the local level weather forecasting, agricultural planning and disaster management.

Reference

1. A. B. Prathyusha and M. S. Prasad, "Design of embedded systems for the automation of drip irrigation," *International Journal of Application or Innovation in Engineering & Management (IJAIEM)*, vol. 1, no. 2, pp. 254–258, Oct. 2012.
2. A. R. Al-Ali, I. Zualkernan, and F. Aloul, "A mobile GPRS-sensors array for air pollution monitoring," *IEEE Sensors Journal*, vol. 10, no. 10, pp. 1666–1671, Oct. 2010.
3. R. Suresh et al., "Intelligent irrigation system—An application of wireless sensor network," *International Journal of Innovative Technology and Exploring Engineering*, vol. 1, no. 2, pp. 112–

115, July 2012.

4. G. D. Abowd et al., "Towards a better understanding of context and context-awareness," in *Proc. Int. Symposium on Handheld and Ubiquitous Computing*, 1999, pp. 304– 307.
5. M. Ramesh and T. Gnanasekaran, "Smart agriculture system using IoT," *International Journal of Engineering Research & Technology*, vol. 6, no. 6, pp. 1–4, 2017.
6. M. Jain and D. Dhama, "IoT based weather forecasting system using Arduino and Raspberry Pi," *International Journal of Computer Sciences and Engineering*, vol. 7, no. 6, pp. 495–499, Jun. 2019.
7. A. Bandyopadhyay and P. Saha, "Weather forecasting using machine learning techniques," in *Proc. 2019 Int. Conf. on Intelligent Computing and Control Systems*, 2019, pp. 782–787.
8. S. O. I. Ameen and M. Shakshuki, "Rainfall prediction based on IoT and machine learning," *Procedia Computer Science*, vol. 170, pp. 34–41, 2020.
9. A. Bhosale and P. Chavan, "IoT based system for rainfall monitoring and alerting," *International Journal of Scientific Research in Computer Science and Engineering*, vol. 6, no. 2, pp. 21–25, 2018.
10. P. Patil and S. Shinde, "Wireless sensor network for precision agriculture," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, no. 5, pp. 216–219, May 2015.
11. A. Kumar and M. Patel, "A real-time weather forecasting and climate monitoring system using IoT," in *Proc. 2020 Int. Conf. on Electronics and Sustainable Communication Systems*, 2020, pp. 694–698.
12. H. Wagh and V. Wankhade, "IoT based weather monitoring and forecasting system using Raspberry Pi," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 6, no. 4, pp. 234–238, Apr. 2017.
13. V. B. Khedkar and S. B. Nandgaonkar, "Design and development of a low-cost IoT based weather monitoring system," *IEEE Int. Conf. on IoT and Applications*, 2017, pp. 1–5.
14. M. Mehmood and M. A. Khan, "A survey on weather forecasting techniques using IoT and machine learning," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 6, pp. 213–220, 2020.
15. H. A. Rahman et al., "Development of an IoT-based weather station," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 12, no. 1, pp. 400–407, Oct. 2018.
16. D. K. Misra et al., "Cloud-based real-time weather monitoring system using Arduino and IoT," *International Journal of Computer Applications*, vol. 179, no. 7, pp. 25– 29, 2018.
17. S. Ghosh and M. Bhowmik, "Forecasting rainfall using machine learning algorithms," *Procedia Computer Science*, vol. 132, pp. 1095–1103, 2018.
18. V. Mishra and N. Bhardwaj, "IoT-based environment monitoring system for smart cities," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 6, no. 4, pp. 3265–3271, Apr. 2017.
19. Adafruit, "BMP280 Barometric Pressure + Temperature Sensor," [Online]. Available: https://learn.adafruit.com
20. Arduino, "Arduino Uno Rev3," [Online]. Available: https://store.arduino.cc