# Raspberry Pi based Weather Prediction and Analysis of Temperature and Humidity using ARIMA Model

Swapna Kulkarni
Assistant Professor, Department of Computer Science, Fergusson College (Autonomous), Pune, India

Amey Bhadane
Department of Computer Science, Fergusson College (Autonomous), Pune, India

Harshiv Gajjar
Department of Computer Science, Fergusson College (Autonomous), Pune, India

**ABSTRACT**

This paper presents a development of weather prediction system for temperature and humidity conditions using DHT11 sensor interfaced to the Raspberry Pi 3B+ board. Data from the DHT11 weather sensor is logged in a spreadsheet file with date and timestamp, analysed for calculating the Heat Index, displayed on the LCD screen, fed to the Autoregressive Integrated Moving Average(ARIMA) model for prediction and then stored to the cloud for IoT applications. The ARIMA model is implemented through python programming which trains the model and predicts future values for a time frame along with specifying the weather conditions like hot-dry, cold-humid, etc. The program also provides accuracy of prediction by analysing the predicted values with the actual recorded values after a set timeframe. The system can be used in various weather prediction applications and can be further improved by adding more weather sensors and increasing the accuracy of the prediction algorithms. The study demonstrates the feasibility and potential of using Raspberry Pi and Python for weather prediction applications.

**Keywords**: ARIMA model, Raspberry Pi, DHT11, IoT, Thing Speak, weather prediction

## 1. INTRODUCTION

Weather forecasting refers to the process of predicting or forecasting the state of the atmosphere at a particular location at a given time in the future. This prediction is based on the collection and analysis of data related to various atmospheric conditions, such as temperature, humidity, dew point, rainfall, and wind speed, using various instruments and technologies like barometers, thermometers, radars, sensors and satellites. The data collected from these sources is used to analyse current weather conditions. Based on this analysis, weather forecasts are made to predict the weather conditions in the future. [1]

Collecting data on atmospheric parameters is only valuable if the data is transmitted quickly and accurately to the users. The transmission and processing of measured data are essential aspects of modern weather forecasting. There are various means to transmit the data, including WI-FI, GSM/GPRS, satellite, and wired links. This ensures that the data is accessible to users such as researchers and meteorologists for timely and accurate weather forecasting. [2]

Weather prediction is necessary because it helps individuals and organizations to prepare and plan for extreme weather conditions such as storms, floods, and heat waves. It also assists in making informed decisions regarding agricultural practices, transportation, and disaster management. Weather prediction also aids in the management of natural resources and the protection of the environment. Additionally, it provides valuable information for businesses and industries, including aviation, energy, and tourism, to plan their operations effectively. Overall, weather prediction is crucial for enhancing public safety, mitigating the effects of extreme weather, and promoting sustainable development.

The proposed3 phase system can be used to provide real-time weather information to individuals, farmers, and businesses, allowing them to plan their activities accordingly. The data collected by the system can also be used to detect weather patterns and forecast extreme weather conditions, which can help to avoid climactic damages and trigger emergency response.Moreover, the system can be integrated with smart agriculture technology to automate farmingtechniques based on the weather conditions or even in smart home applications.

## 2. LITERATURE REVIEW

P.Y. Muck et al. [3] describe a project consisting of three phases aimed at designing and developing a weather station to measure various atmospheric parameters like temperature, humidity, atmospheric pressure, light intensity, wind speed and direction, and rain precipitation. In Phase 1, the weather station prototype is designed and developed using Spark fun Weather Shield sensor and weather meter connected to Arduino Uno R3. In Phase 2, the data collected from the weather sensors is streamed and saved to Google Cloud SQL using Raspberry Pi 3B+ as a gateway. In Phase 3, the project integrates IoT and prediction capabilities, and users can view the weather parameters and predictions in a graphical format through Google Data Studio webpage or an Android mobile application. Real-time notifications are sent to users via social media to help them prepare for future plans. The weather prediction algorithm used is a simple if-else operation.

ARIMA (Autoregressive Integrated Moving Average) is a good model for weather prediction because it uses time series data to predict future observations. Unlike linear regression, which finds a linear relationship between dependent and independent variables, ARIMA takes into account the changing patterns of the data over time. It is a commonly used statistical model for short-term weather forecasting along with other models like Multiple Linear Regression and VAR. ARIMA's ability to consider changes in patterns over time and its use of time series data make it a reliable tool for weather prediction. [4]

The weather forecasting system describes the development of a low-cost environmental monitoring device using the Raspberry Pi 2 as the main platform and inexpensive sensor components such as LM35 analogue temperature sensor, Soil Moisture analogue sensor, Water level detection analogue sensor, and DH11 air temperature and humidity sensor for measuring temperature, soil moisture, water level and humidity. The measured data is transmitted through Wi-Fi to a server provided by data.sparkfun.com, where it is stored for statistical purposes and can be accessed by users globally. The system also includes an MCP 3008 analogue to digital converter, a 16x2 LCD display, and wireless access and 802.11n Wi-Fi adapters for transmitting data wirelessly. [2]

## 3. SYSTEM ARCHITECTURE

Using the available open-source platforms in terms of specification, performance and development tools it was observed the best available device is the Raspberry Pi platform. In this project Raspberry PI 3B+ is used as the main base platform for the project.

For the environment parameter monitoring device, the DHT11 sensor is used for measuring temperature and humidity. The measured parameters are transmitted to the device and then to the server for storing the values which can be periodically used as the statistical data purposes.

In order to transmit the measured data, Wi-Fi is used as the medium through which the data is sent from the Raspberry Pi to the server to be pre-processed, where the user can see the data in the server and download the data and perform additional processing.

The weather forecasting system consists the following hardware components:

- Raspberry Pi3 model B+
- DHT11 air temperature and humidity sensor.
- LCD 16x2 display
- Wi-fi router

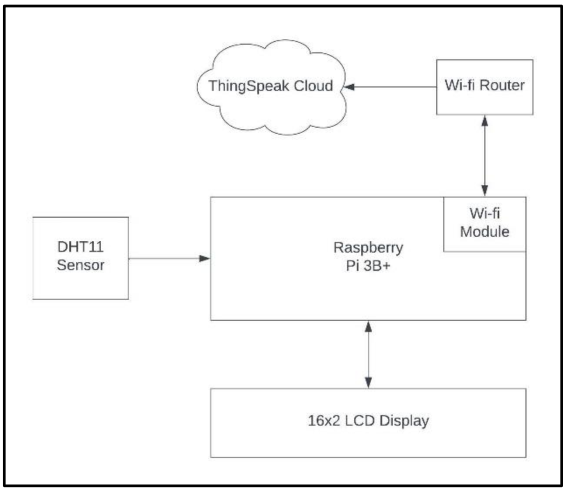



Fig. 1: System Architecture

The software components used in the system are:

- Raspbian OS
- Python Compiler
- Libre Office
- Adafruit library
- Pandas library
- Matplotlib library
- RPi.GPIO module
- Openpyxl module

## 4. METHODOLOGY

The weather prediction system involves three different phases:

Phase I: Weather station design and prototype development.

The weather parameters are measured using DHT11 sensor interfaced to Raspberry Pi 3B+ Board. The measurements taken include temperature, humidity with date and time stamp and stored in a spreadsheet on the local device. The recorded parameters are also displayed on the 16x2 LCD display interfaced to the Pi board.

The data from DHT11 sensor is serial digital data. The temperature range of sensor is $0^0$C to $50^0$C. For measurement of temperature NTC (Negative Temperature Coefficient) thermistor is used. The relative humidity range of DHT11 sensor is 20% to 90% and humidity measurement is capacitive. Operating voltage of DHT11 is +3.5V to +5V. Resolution of DHT11 for temperature and humidity both are 16-bit. Its accuracy for temperature is ±1°C and for humidity is ±1%.Data from sensor consists of decimal and integral parts. A complete data transmission is 40-bit, and the sensor sends higher data bits first as shown below. [5]

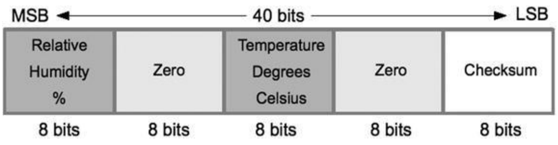



Fig. 2: Data from DHT11 Sensor

The 16x2 LCD display is a commonly used alphanumeric display with 16 columns and 2 rows. It isused to display simple text-based informationsuch as sensor readings, status updates, and user input prompts in projects involving microcontrollers like the Raspberry Pi 3B+. The display communicates with the Raspberry Pi using the GPIO pins. The display has 8-bit data bus which can be used to send data as well as command signals to it. The RS (Register Select) and RW(Read/Write) pins allow to select if data needs to be sent to the command register or the data register and whether to read or write data to the display.


Fig. 3: 16x2 LCD Display

The data from sensors is taken as an input to the Raspberry Pi board. It needs to be displayed onto the LCD display and stored to the local device in a spreadsheet along with a timestamp. This is achieved using a python program which is run on the Raspberry Pi. While storing data to the local device, we also calculate the Heat Index as:

Heat Index = c1 + (c2 * T) + (c3 * R) + (c4 * T * R) + (c5 * T**2) + (c6 * R**2) + (c7 * T**2 * R) + (c8 * T * R**2) + (c9 * T**2 * R**2)

Where, c1 to c9 are constants, T is the temperature in $^{0}$C and R is relative humidity in %. The heat index is the "feels-like" temperature, or how hot it really feels when the relative humidity is factored in with the actual air temperature. [7]

The block diagram of the weather station design using DHT11 sensor and 16x2 LCD display interfaced to the Raspberry Pi 3B+ is:

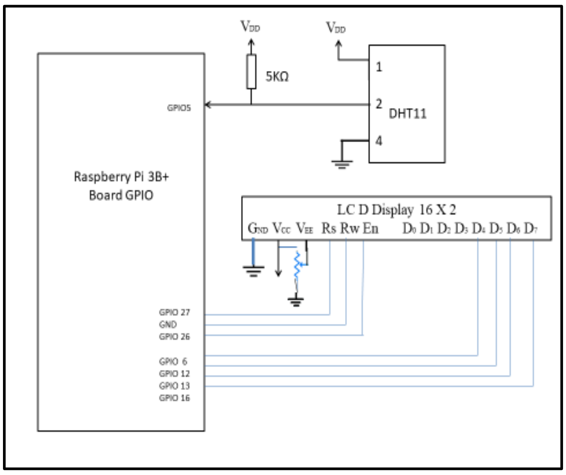



Fig. 4: Block Diagram of weather station

<u>Phase II: Transferring data to initial state cloud platform.</u>

Raspberry Pi 3 Model B+ acts as a gateway between the sensor and IoT platform. It uses the on-board Wi-fi module to connect to the internet to stream and save the data to Cloud platform directly via Wi-Fi. The data is recorded and uploaded to the Cloud using a single python program as mentioned in Phase I. Cloud platform that is used in this project is ThingSpeak. ThingSpeak is an IoT analytics platform service that allows you to aggregate, visualize, and analyse live data streams in the cloud.

<u>Phase III: Weather prediction phase using ARIMA algorithm.</u>

ARIMA is a time series forecasting model that uses three parameters - p, q, and d - to capture the autocorrelation, seasonality, and trend in the time series data."p" is the number of lag observations (or time steps) included in the model. This parameter is used to capture the autocorrelation in the time series data."d" is the degree of differencing required to make the time series stationary. This parameter is used to remove the trend from the time series data."q" is the size of the moving average window. This parameter is used to capture the moving average of the lagged observations.

The weather prediction is implemented using a python program which contains the code to execute the ARIMA(2, 1, 0) model and takes input in the form oftime-stamped data of temperature and humidity parameters whichit uses to train the model. It then predicts the future values of temperature and humidity over a certain timeframe. The output of the program is the predicted values of temperature and humidity and can be visualized using the ‘matplotlib’ library in python. The data is analysed and checked for threshold values to then predict the nature of the weather such as hot-humid, cold-dry, etc.

The ARIMA model combines three different statistical models: autoregressive (AR), integrated (I), and moving average (MA) models. It aims to explain the autocorrelation within the data and can be applied to both stationary and non-stationary data. [6]

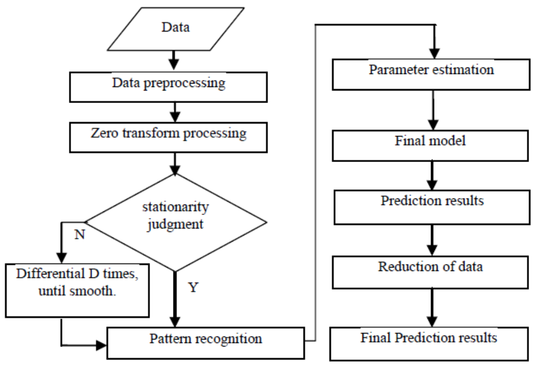


Fig. 5: Representation of ARIMA

The accuracy of prediction is calculated with the help of error rate of prediction which is given by:

$$ErrorRate = \frac{|\ Predicted\ Value - Actual\ Value\ |}{Actual\ Value} \times 100\%$$

The accuracy from error rate is calculated as:

$$Accuracy = 100\% - Error\ Rate$$

The data collected from the sensors is logged onto the local device and after a certain timeframe, the weather is predicted using the previous data collected and compared with the actual data collected from the sensors to evaluate the accuracy of the weather prediction.

5. RESULT

The presented system aims at providing a cost-effective and simple way at analysing and predicting the weather conditions being measured viz. temperature and humidity. The Python program takes input as the recoded timestamped values from the DHT11 sensor and outputs the predicted values for a given timeframe, the expected weather conditions and the accuracy of the prediction algorithm.

The ARIMA model proves to be an accurate method for weather prediction with minimal error rates and whose accuracy will increase over time as more and more data is fed to it for analysis and training. The data is then sent to the cloud for visualisation and further analysis by the user and can be accessed from anywhere on Earth using the internet.

The following figures represent the processes and outputs of the data sensing and data logging process, analysis and prediction of the data using the Python program and data visualisation on the ThingSpeak IoT cloud.

a. Data logging onto the local device in a spreadsheet file:

| Date | Time | Temp(C) | Temp(F) | Humidity(RH) | Heat Index (HI) |
|---|---|---|---|---|---|
| 2023-01-25 | 17:55:46 | 28 | 82.4 | 49 | 28.36 |
| 2023-01-25 | 17:55:52 | 28 | 82.4 | 52 | 28.63 |
| 2023-01-25 | 17:55:57 | 28 | 82.4 | 51 | 28.54 |
| 2023-01-25 | 17:56:03 | 28 | 82.4 | 51 | 28.54 |
| 2023-01-25 | 17:56:09 | 28 | 82.4 | 51 | 28.54 |
| 2023-01-25 | 17:56:14 | 28 | 82.4 | 51 | 28.54 |
| 2023-01-25 | 17:56:22 | 28 | 82.4 | 51 | 28.54 |
| 2023-01-25 | 17:56:28 | 28 | 82.4 | 52 | 28.63 |
| 2023-01-25 | 17:56:34 | 28 | 82.4 | 51 | 28.54 |
| 2023-01-25 | 17:56:39 | 28 | 82.4 | 52 | 28.63 |
| 2023-01-25 | 17:56:45 | 28 | 82.4 | 52 | 28.63 |
| 2023-01-25 | 17:56:51 | 28 | 82.4 | 52 | 28.63 |
| 2023-01-25 | 17:56:56 | 28 | 82.4 | 52 | 28.63 |
| 2023-01-25 | 17:57:02 | 28 | 82.4 | 52 | 28.63 |
| 2023-01-25 | 17:57:08 | 28 | 82.4 | 52 | 28.63 |
| 2023-01-25 | 17:57:13 | 28 | 82.4 | 51 | 28.54 |
| 2023-01-25 | 17:57:19 | 28 | 82.4 | 51 | 28.54 |
| 2023-01-25 | 17:57:25 | 28 | 82.4 | 51 | 28.54 |
| 2023-01-25 | 17:57:30 | 28 | 82.4 | 51 | 28.54 |
| 2023-01-25 | 17:57:39 | 28 | 82.4 | 51 | 28.54 |
| 2023-01-25 | 17:57:44 | 28 | 82.4 | 52 | 28.63 |

Fig. 6: Data logging onto local device

b. Measured parameters displayed on the LCD display:


Fig. 7: Sensed parameters displayed on LCD

c. Data sent and processed on the ThingSpeak Cloud:

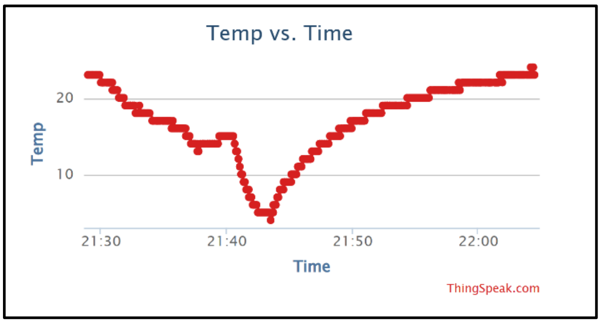



Fig. 8.1: Visualisation of Temperature data on ThingSpeak Cloud

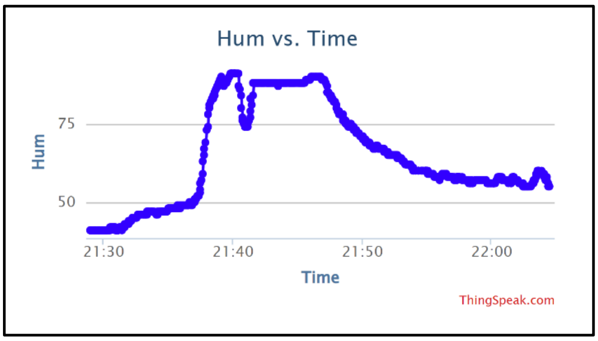
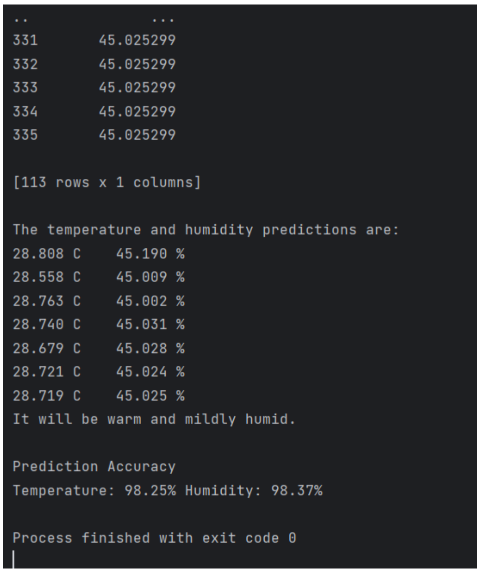



Fig. 8.2: Visualisation of Humidity data on ThingSpeak Cloud

d. Output of predicted values:



```
..            ...
331       45.025299
332       45.025299
333       45.025299
334       45.025299
335       45.025299

[113 rows x 1 columns]

The temperature and humidity predictions are:
28.808 C    45.190 %
28.558 C    45.009 %
28.763 C    45.002 %
28.740 C    45.031 %
28.679 C    45.028 %
28.721 C    45.024 %
28.719 C    45.025 %
It will be warm and mildly humid.

Prediction Accuracy
Temperature: 98.25% Humidity: 98.37%

Process finished with exit code 0
```

Fig. 9: Output of python program for prediction using ARIMA model

## 6. FUTURE SCOPE

More sensors can be added to measure additional weather parameters such as air pressure, wind speed and direction, and UV index. This will enable more accurate weather predictions and allow users to make informed decisions. Features can be added for specialized purposes such as predicting weather conditions for agriculture, aviation, and maritime activities.

The current weather prediction system developed using Raspberry Pi 3B+ and DHT11 sensor can be enhanced by adding more specialized sensors that measure parameters specific to agriculture such as soil pH, atmospheric pressure, solar radiation, wind speed and direction, rainfall, etc. For example, by measuring the soil moisture content and rainfall, the system can determine when to irrigate crops to prevent over-watering or under-watering, which can lead to crop failure. Furthermore, the weather forecasting system can be integrated with other agricultural systems like automatic irrigation systems, pest control systems, and fertilization systems. This will help in creating an automated and optimized agricultural system that can operate with minimal human intervention.

Machine learning algorithms can be integrated into the system to analyse historical weather data and make more accurate predictions. The system can be made more user-friendly by developing a mobile application that allows users to access weather information and receive alerts. Finally, the system can be expanded to cover a larger geographical area by setting up a network of weather stations connected through the internet.

## 7. CONCLUSION

The weather station built using open-source platforms and inexpensive components has proven to be a viable solution for weather prediction. With the ability to transmit data to the cloud and analyse it using the ARIMA model, this system has potential applications in various fields. The three phases of the system are designed to complement each other, resulting in an effective and reliable system. With the possibility of further improvements such as the addition of more sensors and machine learning algorithms, the system can be expanded to have a more significant impact. Overall, this research showcases the potential of open-source technology in creating innovative and accessible solutions for real-world problems which can be easily and widely implemented to grow an interconnected network.

## 8. REFERENCES

[1]Soumitra De, Sayan Das. “LIVE WEATHER PREDICTION USING RASPBERRY PI 3” (2019) International Journal of Advanced Technology in Engineering and Science (IJATES)

[2]K. VivekBabu, K.Anudeep Reddy, CM. Vidhyapathi and B. Karthikeyan. “WEATHER FORECASTING USING RASPBERRY PI WITHINTERNET OF THINGS (IOT)” (2017) ARPN Journal of Engineering and Applied Sciences

[3] P Y Muck and M J Homam.“Iot Based Weather Station Using Raspberry Pi 3” (2018) International Journal of Engineering & Technology

[4] Dharsan G M “A SURVEY ON WEATHER FORECASTING AND THEIR TECHNIQUES” (2022) International Research Journal of Modernization in Engineering Technology and Science

[5] Gay, W.W. “Experimenting with Raspberry Pi.” (2014) A press, Berkeley

[6] Garima Jain, Dr. Bhawna Mallick “A Study of Time Series Models: ARIMA and ETS”

[7] Brian Donegan “What is the Heat Index and Why Is It Used?” (2016) https://weather.com/safety/heat/news/heat-index-feels-like-temperature-summer