Smart Weather Monitoring System Using IoT and AI

Athira R P[1], Hridya Suresh[2], Aswath S[3], Deebu U S[4]
[1,2,3,4]*Department of Electronics and Communication Engineering, College of Engineering Attingal*
doi.org/10.64643/IJIRTV12I10-204540-459

***Abstract*—Accurate weather forecasting is essential for agriculture, disaster management, transportation, and smart city planning. Conventional weather prediction systems primarily depend on large-scale meteorological stations and satellite data, which may not provide precise localized forecasts. To address this limitation, this paper presents a Smart Weather Prediction System using the Internet of Things (IoT) and Artificial Intelligence (AI).**

The proposed system employs IoT-based environmental sensors to continuously monitor parameters such as temperature, humidity, rainfall, and air quality. The collected real-time data is transmitted to a cloud platform for storage and processing. Machine learning algorithms are applied to analyze both historical and real-time datasets to identify patterns and predict short-term weather conditions.

The integration of IoT-enabled sensing with AI driven predictive modeling enhances forecasting accuracy while maintaining low cost and scalability. Experimental results demonstrate improved prediction performance for localized weather conditions compared to conventional observation-based systems. The proposed system can be effectively deployed in agriculture, disaster warning systems, and smart city infrastructures.

I. INTRODUCTION

Weather monitoring plays a crucial role in agriculture, environmental management, disaster preparedness, and urban planning. Accurate real-time information about environmental parameters such as temperature, humidity, rainfall, and air quality help in making informed decisions and reducing potential risks. Traditional weather monitoring systems depend on centralized meteorological stations, which may not provide continuous, location-specific data at a micro level.

With the advancement of the Internet of Things (IoT), environmental parameters can now be monitored in real time using distributed sensor networks. IoT-enabled devices allow continuous data acquisition from sensors that measure temperature, humidity, rainfall, and air quality. These sensors transmit data to cloud platforms for storage and further analysis. However, raw data alone is not sufficient to provide meaningful insights or predictions.

To enhance data interpretation and forecasting capability, Artificial Intelligence (AI) techniques are integrated into the system. AI algorithms analyze historical and real-time environmental data to identify patterns, detect anomalies, and predict short-term weather conditions. By combining IoT based sensing with AI-driven analytics, the proposed Smart Weather Monitoring System provides accurate, real-time, and intelligent weather insights.

II. LITERATURE SURVEY

Rahut et al. [1] proposed a Smart Weather Monitoring and Real-Time Alert System using IoT, which emphasizes timely weather updates and emergency alerts. Their system combines IoT sensors with a cloud-based platform to monitor temperature, humidity, and rainfall. An alert mechanism is incorporated to notify users during extreme weather conditions. The key strength of their system lies in its real-time notification capability, making it highly useful for disaster management and emergency response applications. However, the system does not deeply integrate machine learning models for future weather forecasting.

Mohit Tiwari et al. [2] proposed a weather monitoring system that integrates IoT technology with cloud computing to collect and analyze real-time meteorological data. Their system deploys various environmental sensors to monitor parameters such as temperature, humidity, and atmospheric pressure. The collected data is transmitted to a cloud platform for processing, enabling real-time access, storage, and analysis. Their research highlights the potential of IoT and cloud-based systems in enhancing the accessibility and scalability of weather data. The major advantage of their system is its ability to provide localized and real-time weather insights, which are highly beneficial for agriculture, disaster management, and urban planning.

G.A. Girija C et al. [3] developed an IoT-based weather monitoring system focused on real-time data acquisition and improved forecasting accuracy. Their system integrates multiple sensors to measure key environmental parameters including temperature, humidity, rainfall, and atmospheric pressure. The collected data is processed and visualized through a centralized system, improving monitoring efficiency. Their work demonstrates how IoT-enabled environmental monitoring can enhance data reliability and accessibility. However, their approach primarily focuses on monitoring rather than advanced predictive modeling using AI techniques.

III. METHODOLOGY

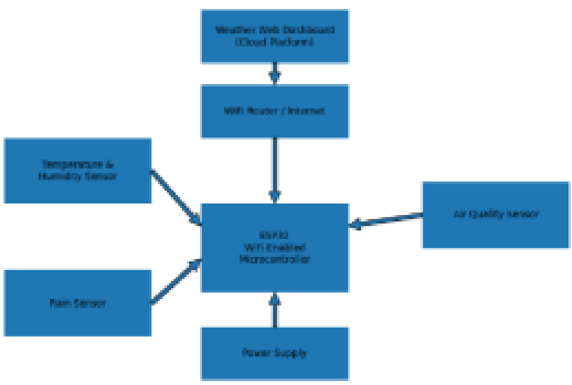

Fig. 3.1: Block Diagram

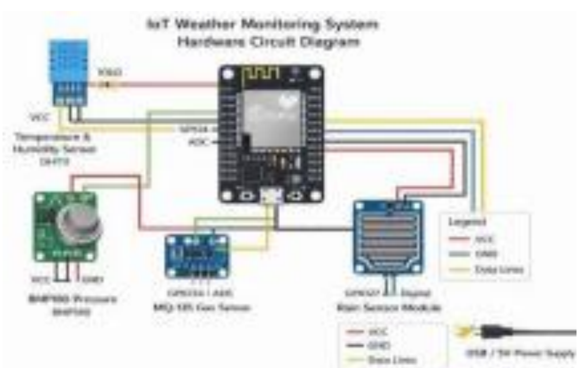

Fig.3.2: Circuit Diagram

Once the environmental data is acquired, the ESP32 performs preliminary processing, including noise reduction, calibration adjustments, and data formatting to ensure measurement accuracy. After processing, the data is transmitted wirelessly through the built-in Wi-Fi module to a cloud-based platform. The cloud server stores the incoming data in a structured database, enabling continuous logging and remote accessibility. A web-based dashboard is integrated with the cloud system to visualize the collected parameters in graphical and numerical formats, allowing users to monitor real time weather conditions from any location.

To enhance the intelligence of the system, the stored dataset undergoes preprocessing techniques such as data cleaning, normalization, and timeseries structuring before being fed into an AI model. The integration of Artificial Intelligence enables the system to analyze historical and real-time environmental patterns. Machine learning algorithms are trained to identify trends in temperature variation, rainfall occurrence, humidity fluctuations, and air quality changes. Based on learned patterns, the system can generate short-term weather predictions and classify environmental conditions.

By combining IoT-based real-time sensing with AI driven predictive analysis, the proposed methodology ensures accurate monitoring, intelligent forecasting, and improved environmental awareness. The system architecture is scalable, cost-effective, and suitable for applications in agriculture, pollution monitoring, disaster preparedness, and smart city infrastructure.

A. SYSTEM ARCHITECTURE

The system architecture consists of three major components: Data Acquisition Layer, Data Processing Layer, and Prediction Layer.

Data Acquisition Layer

This layer includes all hardware components responsible for collecting environmental data. Sensors such as the temperature and humidity sensor, atmospheric pressure sensor, rain sensor, and air pollution sensor continuously monitor the surrounding environment. These sensors generate electrical signals corresponding to the measured environmental parameters.

Data Processing Layer

The ESP32 microcontroller acts as the central processing unit of the IoT system. It receives the sensor signals, processes the raw data, and converts them into digital values. The processed data is then transmitted to the cloud through Wi-Fi communication.

Prediction Layer

In this layer, machine learning algorithms analyze the collected data. Historical weather data is used to train prediction models, and the trained models are used to predict rainfall conditions. Algorithms such as Linear Regression and Random Forest Regression are implemented to improve prediction accuracy.

B. DATA PREPROCESSING

Before applying machine learning algorithms, the collected dataset must undergo preprocessing to improve data quality. The dataset obtained from Kaggle contains several weather attributes that are used for rainfall prediction.

First, the dataset is loaded into the Python environment using the Pandas library. Missing values present in the dataset are handled using mean value replacement to ensure data consistency. Categorical variables are converted into numerical format using dummy encoding techniques.

After cleaning the dataset, the input features and target variable are separated. Environmental parameters such as temperature, humidity, pressure, and air quality are used as input features, while rainfall conditions are used as the target variable.

The dataset is then divided into training and testing sets using the train_test_split() function. Typically, 80% of the data is used for training the machine learning model and 20% is used for testing its performance.

IV. RESULT

To evaluate the performance of the proposed weather prediction system, two machine learning algorithms, namely Linear Regression and Random Forest Regression, were implemented and compared. The dataset was divided into training and testing sets using an 80:20 ratio. The models were evaluated using two performance metrics: Mean Squared Error (MSE) and R^2 score. Mean Squared Error (MSE) measures the average squared difference between the predicted values and the actual values. A lower MSE value indicates better prediction accuracy. The R^2 score, also known as the coefficient of determination, measures how well the model explains the variance in the data. A value closer to 1 indicates a better model performance.

From the results obtained, the Linear Regression model produced a training MSE of 0.04936 and a testing MSE of 0.093822. The corresponding R^2 scores were 0.652177 for training and 0.446371 for testing. In contrast, the Random Forest Regression model produced significantly better results. The training MSE was 0.000001, and the testing MSE was 0.0, which indicates extremely low prediction error. Similarly, the R^2 score for the training dataset was 0.999993, and the testing R^2 score was 1.0, showing that the model explains almost all the variance in the dataset.

1000

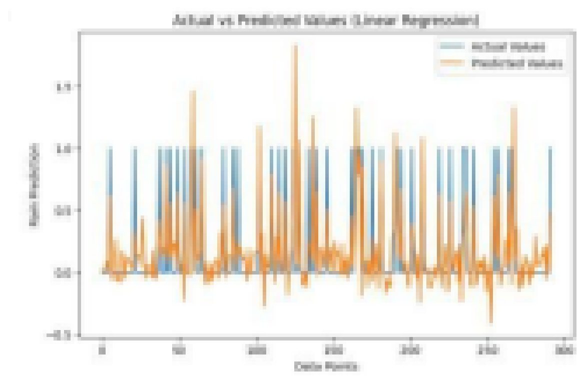

Fig.4.1: Actual Vs Predicted value

From the graph, it can be observed that the predicted values follow the general trend of the actual rainfall data but do not perfectly match all the data points. In some cases, the predicted values deviate from the actual values, indicating prediction errors. This occurs because Linear Regression assumes a linear relationship between input features and the target variable, which may not fully capture the complex patterns present in weather data.

CONFUSION MATRIX ANALYSIS

The confusion matrix contains four important components:

- True Negatives (TN): 58

These represent the number of cases where the model correctly predicted that there would be no rainfall.

- False Positives (FP): 0

These represent cases where the model incorrectly predicted rainfall when there was actually no rainfall.

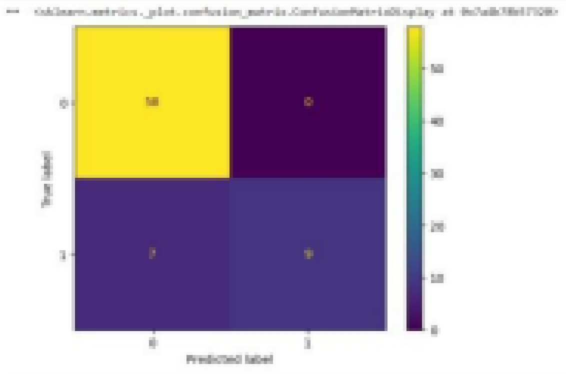

Fig.4.2: Confusion Matrix

- False Negatives (FN): 7

These represent cases where the model predicted no rainfall but rainfall actually occurred.

- True Positives (TP): 9

These represent cases where the model correctly predicted the occurrence of rainfall.

From the confusion matrix, it can be observed that the model performs well in predicting no-rain conditions, as indicated by the high number of true negatives. However, some rainfall events were not correctly predicted, as shown by the false negative

FINAL WEATHER PREDICTION RESULT DISPLAYED ON WEB INTERFACE

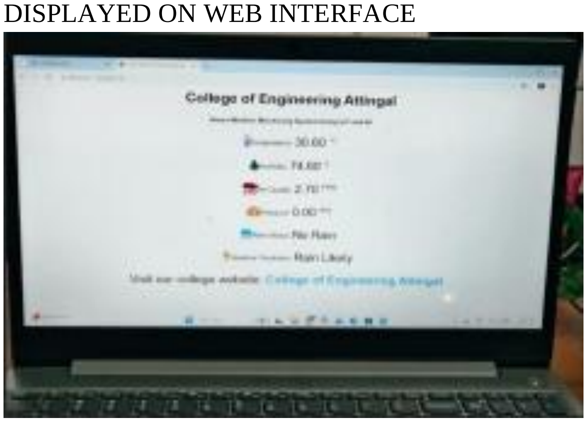

Fig.4.3: Final Output

The final output of the proposed smart weather monitoring and prediction system is displayed through a web-based interface. The system collects environmental parameters such as temperature, humidity, atmospheric pressure, rainfall detection, and air quality using IoT sensors connected to the ESP32 microcontroller. These real-time sensor values are transmitted to the server through a wireless network.

REFERENCES

[1] Mahala, S. K. Jayasingh, and M. R. Kabat, "Weather monitoring and prediction system for the Odisha region," in *Emerging Trends in Computer Science and Engineering*, ch. 40, pp. 509–523, 2022.

[2] M. Tiwari, D. Narang, P. Goel, A. Gadhwal, A. Gupta, and A. Chawla, "A climate observing framework utilizing IoT and cloud computing innovations," International Journal of Innovations in Engineering and Science, vol. 8, no. 7, pp. 23–26, 2023.

[3] C. Girija, A. G. Shires, H. Harshalatha, and H. P. Pushpalatha, "A weather monitoring system based on the Internet of Things (IoT)," International Journal of Engineering Research & Technology (IJERT), vol. 6, no. 13, pp. 1–4, 2018.

[4] Y. Rahut, R. Afreen, and D. Kamini, "A keen IoT-based framework for climate observing and real-time alarms," International Journal of Creative Research Thoughts (IJCRT), vol. 12, no. 2, pp. e876–e885, 2023, doi: 10.5120/ijcrt2402571.

[5] K. Ashton, "That 'Internet of Things' thing," RFID Journal, vol. 22, no. 7, pp. 97–114, 2009.

[6] D. Giusto, A. Iera, G. Morabito, and L. Atzori, The Internet of Things: 20th Tyrrhenian Workshop on Digital Communications. New York, NY, USA: Springer, 2010.

[7] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," IEEE Internet of Things Journal, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[8] S. R. Nandhini and S. Porkodi, "Weather prediction using machine learning techniques," International Journal of Advanced Research in Computer Science, vol. 8, no. 5, pp. 123–126, 2017.

[9] R. K. Gupta and P. K. Yadav, "Weather forecasting using machine learning algorithms," International Journal of Computer Applications, vol. 179, no. 32, pp. 1–5, 2018.

[10] L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
[11] T. M. Mitchell, Machine Learning. New York, NY, USA: McGraw-Hill Education, 1997.
[12] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
[13] J. Brownlee, Machine Learning Mastery with Python. Machine Learning Mastery, 2016.
[14] Kaggle, “Weather dataset for prediction,” [Online]. Available: Kaggle. Accessed: 2026.