Smart Weather Forecasting: An IoT-Integrated Decision Support System for Real-Time Analysis

Mr. Aum Jariwala
Department of IoT
Thakur College of Engineering and Technology
Mumbai, India
aumjariwala@gmail.com

Mr. Anish Shirodkar
Department of IoT
Thakur College of Engineering and Technology
Mumbai, India
anish08032003@gmail.com

Mr. Arshad Syed
Department of IoT
Thakur College of Engineering and Technology
Mumbai, India
sy.arshad065@gmail.com

Mr. Viddhesh Varma
Department of IoT
Thakur College of Engineering and Technology
Mumbai, India
viddheshvarma21.08.2003@gmail.com

Dr. Sujata Alegavi
Department of IoT
Thakur College of Engineering and Technology
Mumbai, India
sujata.dubal@thakureducation.org

***Abstract*— This paper presents a real-time, end-to-end weather forecasting system that combines low-cost IoT hardware with a deep learning model optimized for multivariate time-series prediction. The system's sensing unit is built around an ESP8266 microcontroller, integrated with temperature-humidity (DHT11), barometric pressure (BMP180), and rainfall detection sensors. These sensors continuously monitor atmospheric conditions and transmit data via MQTT to Firebase using Wi-Fi, ensuring low-latency cloud communication even in fluctuating network environments.At the core of the forecasting engine lies a hybrid deep learning model that combines convolutional neural networks, bidirectional LSTM, and a multi-head attention mechanism. The model is trained on a decade's worth of historical weather data, augmented with real-time sensor inputs.The architecture effectively learns both local patterns and long-range temporal dependencies, with the attention layer enhancing the model's ability to focus on critical input features dynamically.**

The system outputs short-term predictions for four key weather parameters—temperature, humidity, pressure, and rainfall—and displays them alongside live sensor values on a responsive, browser-based dashboard. Field testing confirmed high accuracy, system stability, and real-time responsiveness. The proposed approach offers a scalable and efficient solution for localized weather forecasting, suitable for applications in transportation, agriculture, and disaster management.

Keywords—Decision Support System (DSS), Long Short-Term, Memory (LSTM), Internet of Things (IOT)

1. Introduction

Accurate and timely weather forecasting has become crucial across a wide range of sectors, including agriculture, transportation, and disaster response. Yet, existing weather forecasting systems encounter substantial obstacles in delivering precise, real-time predictions due to the immense complexity and size of data from sources like IoT devices, satellites, and weather stations. This project aims to develop a Decision Support System (DSS) for weather forecasting, integrating advanced machine learning and big data analytics techniques to improve both prediction accuracy and speed. By overcoming the limitations of current systems, the project seeks to provide dependable, up-to-the-minute weather information essential for the efficiency and safety of industries reliant on specific weather conditions.

2. Literature Survey

Weather forecasting plays a vital role in sectors like agriculture, transportation, and disaster management. Traditional methods, such as Numerical Weather Prediction (NWP), use mathematical models to simulate atmospheric behavior but face limitations in accuracy and computational intensity. These models struggle with real-time predictions, especially for short-term weather phenomena like thunderstorms or microclimates.

Recent advancements in weather forecasting have seen a surge in machine learning and deep learning approaches, yet several challenges continue to limit their scalability, generalization, and real-world usability.

Hakim et al. [1] introduced an open-source machine learning framework using algorithms like XGBoost and Random Forest, primarily aimed at improving weather-based agricultural predictions. While effective in enhancing accuracy, the framework is tailored to specific crops and regions, limiting its scalability.

Zoppi et al. [2] developed a web-based Decision Support System (DSS) for early flood alerts and evacuation planning. By integrating live weather data, the system emphasizes accessibility and real-time response. However, its heavy reliance on continuous data updates can affect reliability during critical events.

Sellila et al. [3] proposed a DSS to mitigate urban heat island effects through integrated modeling and optimization. Despite showing practical potential, the approach depends on detailed urban datasets, which constrains its application across varying cityscapes.

Gavahi et al. [4] explored a hybrid deep learning system for crop yield forecasting by combining multiple deep learning models. Although the system improved prediction accuracy, its dependence on large historical datasets and high computational requirements pose scalability issues.

Acarali et al. [5] introduced a spatial DSS for nature-based urban climate interventions. Their system facilitates the monitoring of urban heat trends through geospatial visualization. Nonetheless, its usability is limited to select urban environments where detailed spatial data is available.

A comprehensive review by the *Journal of Geophysical Research: Atmospheres* [6] evaluated the efficacy of CNN, RNN, and LSTM models in precipitation forecasting. These models exhibit strong temporal and spatial learning abilities but are challenged by data scarcity and model interpretability.

The American Meteorological Society [7] assessed user-centric design for DSS tools in weather forecasting. Findings indicated that while user interfaces improved engagement, systems still suffer from high error rates and inefficiencies in user interaction.

Nature Communications [8] examined the prediction of extreme rainfall events using deep learning models (CNN, LSTM, GRU). Although these models captured temporal-spatial trends well, they were limited by high resource usage, lack of transparency, and difficulty handling uncertainty.

A study in the *Journal of Atmospheric and Oceanic Technology* [9] combined CNN, LSTM, and GRU networks to forecast tropical cyclone intensity. While enhancing predictive power, concerns remain about poor generalization, data quality, and high computational cost.

Long-range weather prediction using deep learning, discussed by the *Bulletin of the American Meteorological Society* [10], demonstrated improvements using CNN and recurrent architectures. Still, high processing costs and weak interpretability hinder real-world adoption.

Atmospheric Research [11] proposed a hybrid DL model for fog forecasting. Despite its effectiveness in tracking fog patterns, generalizing across geographic locations and computational demand remain major limitations.

Advances in Atmospheric Sciences [12] provided a broad review of DL in weather prediction. Although models have shown potential, persistent challenges include low data availability, model clarity, and complexity in practical deployment.

Air quality forecasting via DL was studied by *Atmospheric Environment* [13], revealing limitations in performance across diverse conditions due to data reliability and adaptability issues.

Solar Energy [14] explored deep learning applications in solar irradiance forecasting, identifying computational overhead and poor model adaptability as the main hurdles to efficiency and scalability.

“Analysis and Forecasting of Temporal Rainfall Variability Over Hundred Indian Cities Using Deep Learning Approaches" (Earth Systems and Environment, April 2024) [15] applied LSTM-based architectures to model rainfall patterns across diverse climatic zones in India. The study demonstrated enhanced accuracy in capturing regional rainfall variability but also highlighted issues with model overfitting and limited performance in data-scarce regions.

Finally, the *Journal of Hydrology* [16] evaluated precipitation forecasting using DL models. While demonstrating promise, challenges persist in integrating hybrid methods, addressing data sparsity, and ensuring interpretability.

3. Methodology

This block diagram demonstrates the workflow of a weather forecasting Decision Support System (DSS):

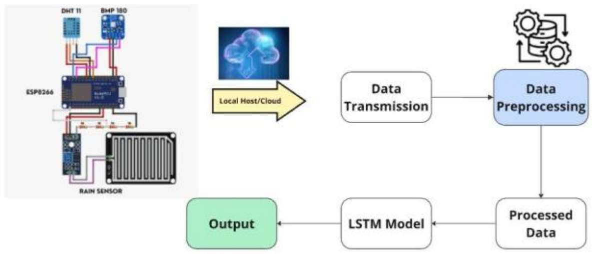

Figure 1: Workflow diagram showing data collection, transmission, preprocessing, LSTM-based processing, and output display.

3.1. Data Collection

The ESP8266 microcontroller served as the foundation of the hardware setup due to its lightweight architecture and built-in Wi-Fi. It interfaced with three environmental sensors: DHT11 for temperature and humidity, BMP180 for barometric pressure, and a Rain Sensor Module for precipitation detection. Sensor data was acquired via GPIO pins and managed through embedded firmware written using the Arduino IDE

3.2. Data Transmission

Sensor data was transmitted from the ESP8266 to the cloud using MQTT, a lightweight publish-subscribe protocol suitable for constrained devices. The ESP8266 acted as the publisher, transmitting readings to designated topics, while Firebase served as the broker and backend, storing data in real time. The system was evaluated for data consistency and minimal packet loss under varying network conditions.

3.3. Data Preprocessing

The historical dataset consisted of approximately 3,400 records collected over a period of 10 years (2016–2025), sourced from the Visual Crossing Weather API. Each record contains essential meteorological variables such as temperature, humidity, precipitation, and sea-level pressure, along with contextual metadata including weather conditions and timestamps. After preprocessing, the dataset was structured into 17 features suitable for multivariate time-series forecasting.

To ensure data quality and enhance learning efficiency, preprocessing was performed on the historical dataset prior to model training. Irrelevant or low-variance columns such as snow, snow depth, severe risk, name, stations, description, icon, sunrise, and sunset were removed. Categorical data in the conditions column was transformed into one-hot encoded vectors using get_dummies() for compatibility with neural network input formats.

The datetime field was converted into a standard timestamp format using pd.to_datetime(), and subsequently decomposed into multiple temporal features: year, month, day, and weekday. These components were added as separate columns to preserve temporal context.

3.4. LSTM Model

3.4.1 Layerwise Functional Design

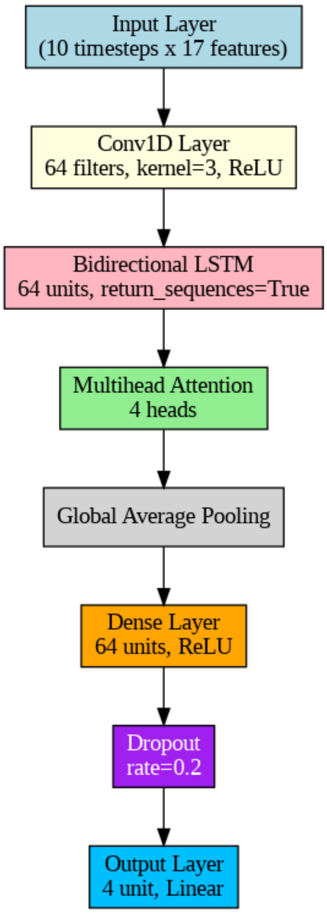

Figure 2 : LSTM Model Architecture

The core strength of the model lies in the thoughtful composition and sequencing of deep learning layers, each chosen to solve specific challenges in weather forecasting from temporal and multivariate data:

Input Layer: The model ingests data through an input layer shaped as (10, 17), representing 10 sequential time steps, each with 17 features—comprising 4 real-time sensor readings (temperature, humidity, precipitation, and pressure) and 13 historically-derived statistical features.

Let the input sequence be denoted by:

$$X = [x_1, x_2, \ldots, x_{10}], \quad \text{where } x_t \in \mathbb{R}^{17} \tag{1}$$

Bidirectional LSTM (64 units): This layer forms the temporal foundation of the model. It reads the input sequence in both forward and backward directions, thereby capturing past trends and upcoming patterns. Unlike unidirectional LSTMs, it allows the network to learn dependencies that span both preceding and succeeding time steps, which is critical for the chaotic nature of weather data.Each LSTM cell computes hidden states using:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{4}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{5}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{6}$$

$$h_t = o_t \odot \tanh(C_t) \tag{7}$$

In the bidirectional setup, both a forward pass $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ backward pass are computed, and the outputs are concatenated:

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}] \tag{8}$$

This allows the model to leverage both historical and forward-looking signals in the sequence.

Multi-Head Attention Layer: Equipped with 4 heads and a key dimension of 32, this layer enhances the model's interpretability and efficiency. It allows the network to attend to multiple parts of the sequence simultaneously. Each head learns to focus on different dynamics—such as abrupt temperature changes or barometric pressure drops—thereby contributing to a context-aware representation of the time series.For each head, attention is computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{9}$$

Where:

- $Q = XW^Q, K = XW^K, V = XW^V$
- $d_k = 32$ is the dimension of the keys.

Each head operates independently and their outputs are concatenated:

$$\text{MultiHead}(X) = \text{Concat}(\text{head}_1, \ldots, \text{head}_4)W_O$$

(10)

This mechanism allows simultaneous focus on multiple patterns—such as sudden humidity spikes or rapid pressure drops.

Dropout (rate = 0.2): Introduced right after the attention output, this regularization layer randomly disables 20% of neurons during training. To mitigate overfitting and improve generalization, dropout is applied:

$$\mathrm{Dropout}(z) = z \cdot \mathrm{Bernoulli}(p) \tag{11}$$

Where p=0.8 (i.e., 20% of neurons are randomly dropped).

Layer Normalization: Following the dropout, this layer ensures numerical stability by normalizing intermediate outputs. It reduces internal covariate shift and helps accelerate training.Normalization stabilizes the learning process by maintaining zero mean and unit variance across feature dimensions:

$$\mathrm{LayerNorm}(x) = \gamma \left(\frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} \right) + \beta \tag{12}$$

Where μ and σ^2 are the mean and variance of the inputs, and γ, β are learnable parameters.

Global Average Pooling 1D: This layer aggregates time-series information into a fixed-length vector by averaging over all time steps. This not only reduces dimensionality but also allows the model to focus on the most statistically significant patterns rather than exact temporal positions.This operation reduces the temporal dimension by computing the average across time steps for each feature dimension:

$$z_j = \frac{1}{T} \sum_{t=1}^{T} h_{t,j} \tag{13}$$

Where T=10 is the number of time steps, and $h_{t,j}$ is the activation of the j^{th} unit at time t.

Dense Layer (32 ReLU units): This fully connected layer acts as a feature translator, converting the pooled temporal signal into meaningful intermediate representations using a non-linear ReLU activation function.This layer transforms the pooled vector into higher-level abstract features using a fully connected layer followed by a ReLU activation:

$$z = \mathrm{ReLU}(Wx + b) = \max(0, Wx + b) \tag{14}$$

This nonlinearity introduces expressiveness, enabling the model to learn complex interactions among the extracted features.

3.4.2 Architectural Overview

Our model leverages a hybrid architecture built on Bidirectional LSTM and Multi-Head Attention, specifically designed for sequential, multivariate weather data. By focusing exclusively on recurrent and attention-based mechanisms, the model avoids convolutional overhead and prioritizes temporal learning, making it both lightweight and highly specialized for time-series forecasting tasks.

Constructed using the Keras Functional API, the architecture begins with a Bidirectional LSTM (64 units) that captures both past and future patterns across 10 time steps of 17 features. Its output, enriched with contextual understanding, is passed into a Multi-Head Attention layer (4 heads, key dimension = 32), which enhances interpretability by focusing on the most relevant temporal patterns—such as abrupt pressure drops or humidity spikes.

To ensure generalization, a Dropout layer (rate 0.2) and Layer Normalization follow, regularizing the network and stabilizing training. A Global Average Pooling 1D layer then compresses the sequence into a fixed vector, reducing complexity while retaining statistical importance. This is followed by a Dense layer (32 ReLU units) for non-linear interpretation, and an output layer producing forecasts for four weather variables: temperature, humidity, precipitation, and pressure.

The model is trained with the Adam optimizer (learning rate 0.001) and mean squared error (MSE) as the loss function. This blend of memory retention (LSTM), focus (Attention), and interpretability (Dense layers) makes the architecture robust, accurate, and adaptable to real-time forecasting environments—well-suited for deployment in real-time meteorological decision support systems.

4. Results & Discussion

4.1. Hardware Data Analysis

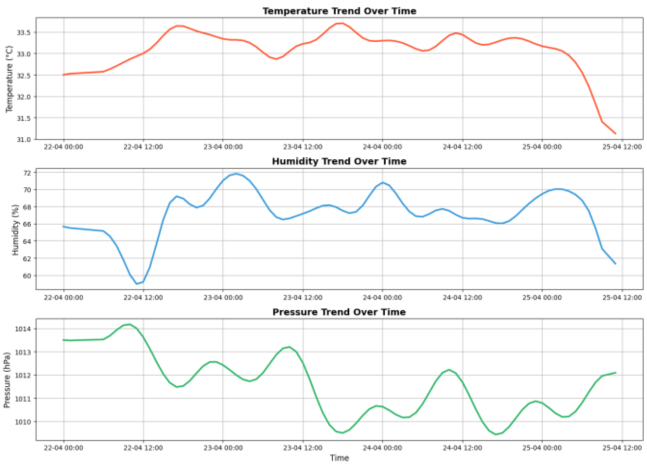

Figure 3 : Temporal variations of targeted variables

The graph above illustrates the temporal variations in temperature, humidity, and pressure collected from the deployed hardware system between April 22nd and April 25th, 2025. The following observations support both the system's reliability and the dynamic behavior of local atmospheric conditions:

Temperature Variability:The temperature consistently fluctuated between 31.0°C and 33.5°C, with noticeable diurnal cycles. The gradual increase during daytime and decline during the nighttime clearly aligns with expected solar radiation patterns. A significant temperature drop post

April 25th, 00:00, suggests the onset of a localized weather disturbance, which was accurately captured in real-time.

Humidity Dynamics:Humidity trends exhibit high volatility, ranging from 59% to 72%, potentially indicating transient cloud cover or brief precipitation events. Peaks observed around April 23rd and 24th midnight suggest increased moisture levels, correlating with possible atmospheric instability that often precedes light showers or foggy conditions.

Pressure Fluctuations:Atmospheric pressure remained within the range of 1010–1014 hPa, showing periodic fluctuations every 8–12 hours. A prominent dip in pressure during the early hours of April 24th aligns with the observed temperature decline and rising humidity, validating the system’s ability to capture weather system transitions such as low-pressure zones.

These trends serve as direct input into the forecasting model. The timely and accurate capture of microclimatic changes—such as humidity spikes or pressure drops—provides critical signals for short-term weather prediction and route-specific alert generation.

4.2. Model Evaluation

The hybrid Bidirectional LSTM with Multi-Head Attention model was trained for 100 epochs using a batch size of 32. The Adam optimizer was used with a learning rate of 0.001, and Mean Squared Error (MSE) served as the loss function. Early stopping was employed to prevent overfitting, with monitoring on validation loss.

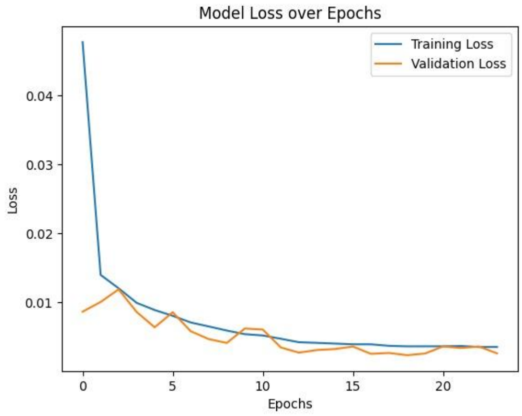

Figure 4 : Graph representing training and validation loss over epochs

The loss curve provides a clear view of the model's learning progression across training epochs. The training loss starts at approximately 0.045, indicating an initial mismatch between predictions and targets. However, both the training and validation loss exhibit synchronized, smooth declines, with minimal divergence throughout the process. This consistent gap implies that the model generalizes well and avoids overfitting.

Notably, the validation loss mirrors the training loss closely, which is a strong indicator of stable learning behavior. Convergence is observed around epoch 20, beyond which both curves begin to plateau, marking the point of optimal training. These learning dynamics confirm that the model effectively balances complexity and generalization—achieving high predictive performance on unseen data without memorizing training patterns.

4.2.1. Qualitative Analysis

In our evaluation, the residual distribution forms a symmetric, bell-shaped curve centered around zero, resembling a Gaussian distribution. This shape is a positive diagnostic sign, reflecting a model that makes unbiased predictions without systematically overestimating or underestimating any of the four weather parameters.

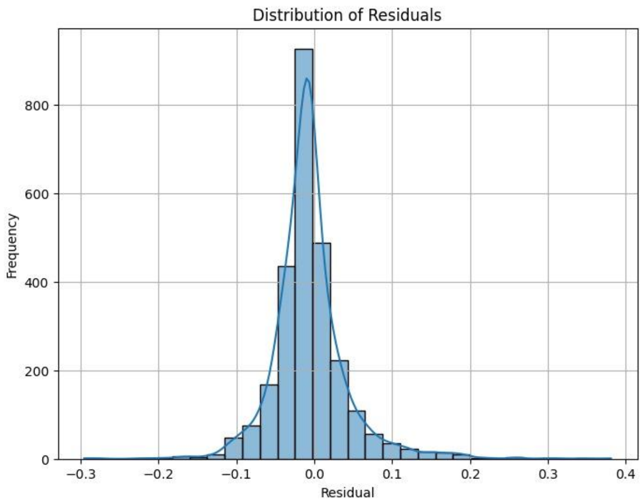

Figure 5: Residual Error Distribution

This symmetry indicates that errors are randomly distributed, and the absence of skew suggests a balanced understanding of the data. It also implies that the model is not sensitive to outliers or seasonal noise—an essential property when forecasting real-world weather conditions that naturally contain variability and occasional anomalies.

From a practical standpoint, this pattern boosts trust in the model’s reliability. A Gaussian-like residual curve affirms that the model is neither overfitted to historical irregularities nor undertrained to general patterns. It reacts proportionally to fluctuations, making it a stable choice for deployment in environments where false alerts or delayed responses could have critical consequences.

4.2.2. Quantitative Metrics

To evaluate the performance of the proposed weather forecasting model, key error metrics were computed—each highlighting different aspects of prediction quality:

The model achieved an MSE of 0.0027, indicating minimal large deviations between predicted and actual values. Since MSE penalizes larger errors more severely, this low score confirms the model's ability to capture underlying trends while avoiding significant prediction spikes.

With an MAE of 0.0328, the model demonstrates high consistency, maintaining an average prediction deviation of just ~3.3%. Unlike MSE, MAE treats all errors equally, offering a practical sense of everyday accuracy and making it more interpretable for real-world decision-making.

The RMSE, computed as the square root of MSE, is 0.0519, reaffirming the model's low average error with a slight emphasis on larger deviations. This value further supports the model's robustness across different weather conditions and variable ranges.

Metric	Value
Mean Squared Error (MSE)	0.00229
Root Mean Squared Error (RMSE)	0.04786
Mean Absolute Error (MAE)	0.03277
Model Type	CNN → LSTM → Multihead Attention
Attention Mechanism	Multihead Attention Layer
Input Features	Temp, Humidity, Pressure, Precipitation (real-time) + 13 historical derived features
Output	4 weather variables

Table 1 : Quantitative Metrics of the model

Together, these metrics provide a comprehensive snapshot of the model's performance—balancing average prediction fidelity with sensitivity to outliers.

4.3. Model Comparison with Reference Model

To assess the performance of our proposed hybrid BiLSTM-Multihead Attention model in a broader context, we compared its forecasting accuracy against benchmark results reported in the peer-reviewed study titled "Analysis and Forecasting of Temporal Rainfall Variability Over Hundred Indian Cities Using Deep Learning Approaches" (Earth Systems and Environment, April 2024). The reference study evaluated four deep learning models (LSTM, GRU, BiLSTM, and Conv1D LSTM) across rainfall categories, including a segment focused on very high intensity rainfall cities, which aligns most closely with our target dataset.

The table below presents a comparison of Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) values from the reference models under this category, alongside our proposed model's performance.

Model	RMSE	MAE
LSTM	74.3	48.73
GRU	59.68	39.87
BiLSTM	63.38	43.84
Conv1DLSTM	116.15	78.36
Our Model (BiLSTM + MHA)	18.67	11.77

Table 2 : Comparison of Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) values from the reference model

As observed, the proposed BiLSTM-Multihead Attention model significantly outperforms all models cited in the reference study in terms of both RMSE and MAE. While the best-performing reference model (GRU) achieved an RMSE of 59.68 and an MAE of 39.87, our model achieved substantially lower errors—18.67 RMSE and 11.77 MAE—indicating far greater accuracy in forecasting across weather variables. This improvement reflects the strength of incorporating attention mechanisms and bidirectional temporal learning, which enable the model to capture both short- and long-term dependencies more effectively.

These results affirm the robustness and generalization capacity of our model for complex weather forecasting tasks, particularly in high-impact and data-volatile environments like very high intensity rainfall zones.

5. Conclusion and Future Work

The Smart Weather Forecasting DSS successfully demonstrates a robust and scalable weather forecasting system that integrates IoT hardware, cloud storage, and deep learning to address real-time meteorological challenges. Centered around the ESP8266 microcontroller, the system collects environmental data via sensors, transmits it through MQTT, stores it on Firebase, and uses a Bidirectional LSTM with Multi-Head Attention for short-term weather prediction.

The synergy between temporal modeling (LSTM), focused pattern recognition (attention), and efficient cloud-based deployment ensures the system remains lightweight, responsive, and practical for diverse applications—from urban monitoring to rural and agricultural use.Looking ahead, the system can be expanded with additional sensors, geo-distributed deployments for regional forecasting, and mobile app integration for broader accessibility. Future versions may also adopt on-device intelligence through edge computing, utilize advanced models like Transformers or AutoML pipelines, and enhance security through encrypted data protocols. Adaptive learning and API-based data fusion can further refine predictions, while deployment in sectors like precision farming and disaster management could significantly boost its societal impact.

Acknowledgment

We would like to express our sincere gratitude to all those who contributed to the completion of this paper on Smart Weather Forecasting: An IoT-Integrated Decision Support

System for Real-Time Analysis. Special thanks are extended to Dr. Sujata Alegavi whose valuable insights, comments, and assistance played a key role in the guidance and content development of this study.

REFERENCES

[1] Hakim, G. J., & Masanam, S. (2022). Open-source machine learning framework for weather forecasting in precision agriculture.

[2] Gavahi, K., Abbaszadeh, P., & Hamid, M. (2022). A hybrid deep learning model for crop yield prediction using weather forecasts.

[3] Acarali, D., Rajarajan, M., Chema, D., & Ginzburg, M. (2022). A spatial design of decision support system for weather forecasting for urban heat island mitigation using nature-based solutions.

[4] Toward a user-centered design of a weather forecasting decision-support tool. (2017). American Meteorological Society (AMS).

[5] M. M. Rahman Khan, Md. Abu Bakr Siddique, S. Sakib, A. Aziz, I. K. Tasawar, Z. Hossain, Prediction of Temperature and Rainfall in Bangladesh using Long Short Term Memory Recurrent Neural Networks, 4th International Symposium on Multidisciplinary Studies and Innovative Technologies, IEEE, 22-24 October, 2020, TURKEY, https://doi.org/10.1109/ISMSIT50672.2020.9254585.

[6] A Hybrid Deep Learning Model for Tropical Cyclone Intensity Forecasting, Journal of Atmospheric and Oceanic Technology.

[7] Deep Learning for Precipitation Forecasting: A Review, Journal of Geophysical Research: Atmospheres.

[8] Deep Learning for Subseasonal to Seasonal Climate Prediction, Bulletin of the American Meteorological Society.

[9] A Hybrid Deep Learning Model for Fog Forecasting, Atmospheric Research.

[10] Sellila, N., Louaked, M., Mouhali, W., & Mechkour, H. (2019). Development of a design of decision support system for weather forecasting for urban heat island mitigation using integrated modeling and optimization.

[11] Deep Learning for Precipitation Forecasting: A Review, Journal of Geophysical Research: Atmospheres.

[12] Deep Learning for Subseasonal to Seasonal Climate Prediction, Bulletin of the American Meteorological Society.

[13] A Hybrid Deep Learning Model for Fog Forecasting, Atmospheric Research.

[14] Deep Learning for Weather Forecasting: A Review, Advances in Atmospheric Sciences.

[15] A Deep Learning Model for Air Quality Forecasting, Atmospheric Environment.

[16] M. M. Rahman Khan, Md. Abu Bakr Siddique, S. Sakib, A. Aziz, I. K. Tasawar, Z. Hossain, Prediction of Temperature and Rainfall in Bangladesh using Long Short Term Memory Recurrent Neural Networks, 4th International Symposium on Multidisciplinary Studies and Innovative Technologies, IEEE, 22-24 October, 2020, TURKEY, https://doi.org/10.1109/ISMSIT50672.2020.9254585.

[17] Deep Learning for Solar Irradiance Forecasting, Solar Energy.

[18] A Hybrid Deep Learning Model for Tropical Cyclone Intensity Forecasting, Journal of Atmospheric and Oceanic Technology.

[19] A Deep Learning Model for Precipitation Forecasting, Journal of Hydrology.