Haldia Institute of Technology Publishing International Journal of HIT Transaction on ECCN
Int.J.HIT.TRANSC:ECCN.Vol.13: Special Issue NSEFCCIC (2026) Page 14-18 ISSN: 0973-6875
Available Online at www.hithaldia.in/locate/ECCN

ORIGINAL CONTRIBUTION

Design and Implementation of an Intelligent Machine Learning-Based Weather Prediction Application

[1]Dr.Sagarika Das, [2]Nikita Shaw, [3]Nabarun Naskar, [4]Mith Biswas, [5]Md. Raquif, [6]Mayukh Santra

[1]*Faculty , Dept.of Electronics and Communication Engineering, Haldia Institute of Technology, Haldia, Purba Medinipur, West Bengal*

[2,3,4,5,6]*UG student, Dept. of Electronics and Communication Engineering , Haldia Institute of Technology, Haldia, Purba Medinipur, West Bengal*

ABSTRACT

Accurate weather forecasting is essential for agriculture, transportation, disaster management, and environmental planning. This study presents a comparative weather forecasting model using three time-series and deep learning approaches: Autoregressive Integrated Moving Average (ARIMA), Seasonal ARIMA (SARIMA), and Long Short-Term Memory (LSTM). Historical weather data containing variables such as temperature, humidity, atmospheric pressure, and rainfall are preprocessed to handle missing values, normalize the data, and identify temporal patterns. ARIMA is applied to capture linear trends and short-term dependencies, while SARIMA incorporates seasonal variations in weather patterns. LSTM, a recurrent neural network architecture, is employed to learn complex nonlinear relationships and long-term temporal dependencies. The models are trained and evaluated using appropriate error metrics, including Mean, Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). The comparative analysis identifies the most effective approach for accurate weather prediction and demonstrates the potential of machine learning and deep learning techniques for reliable forecasting.

KEYWORDS: ARIMA, SARIMA, LSTM, Time Series Forecasting, Deep Learning, Machine Learning.

1. INTRODUCTION

Weather forecasting is one of the most important as well as a challenging task in the field of science and technology in daily life. Weather conditions such as temperature, rainfall, humidity, wind speed, and pressure directly affect human activities, agriculture, transportation, disaster management, aviation, marine operations, and energy consumption. Due to these variations, predicting future weather accurately is not easy. In many cases, people depend on general weather reports from websites, mobile applications, or news channels. However, for academic, local, and research-based applications, it is useful to build a system that can analyze available weather data and generate predictions automatically.In this project, a smart weather forecasting system is developed using statistical models like ARIMA, SARIMA, and LTSM which collects and uses previous weather data for forecasting future weather. RMSE is used to compare all the models and generate lowest error. This project is useful for understanding the basic concept of weather forecasting using AI/ML. The current scope of the project includes forecasting temperature for the next 10 days. However, the present system is limited to temperature prediction only. It does not forecast other weather parameters such as rainfall, humidity, wind speed, pressure, or cloud condition. In the future, the project can be extended by adding more weather parameters and using a larger dataset. It can also be improved by using real-time weather data from sensors pr online weather APIs.

2. METHODOLOGY

The methodology of this project explains step-by-step procedure to develop the smart weather forecasting system. This project uses AI/ML based models such as ARIMA, SARIMA, LSTM etc. The methodology of this project is extended by adding an application layer after model training and prediction.

2.1 System Architecture:

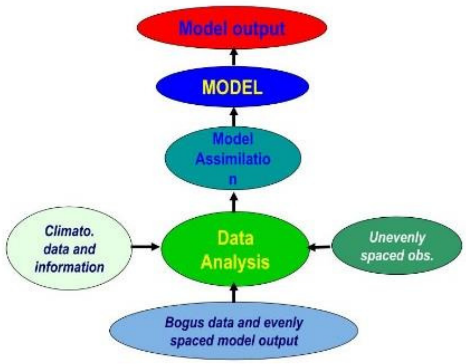

Figure 1: ML and Weather Forecasting Flowchart

The system architecture starts with input weather records and ends with final forecast output. This system reads the data set and automatically detects the date-time column and temperature column. After loading the dataset, the date-time column is converted into proper date-time format. The hourly temperature data is then converted into daily average temperature data.

The system uses three forecasting models: ARIMA, SARIMA, and LSTM. ARIMA is used to identify general trends in the temperature data. SARIMA is used to identify both trend and seasonal patterns. LSTM is used as a deep learning model to learn long-term patterns from previous temperature values. After training all models generates prediction which is evaluated using RMSE. It reduces the prediction error. The model with lowest RMSE is selected as the best model.

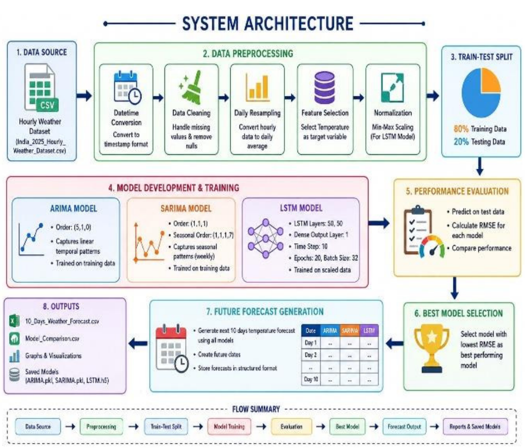

Figure 2: System Architecture Flowchart

2.2 Data Description

The quality and reliability of the weather forecasting depend heavily on the availability on the historical weather observations. In the given code, the dataset file named India_2025_Hourly_Weather_Dataset.csv. is loaded using the Pandas library.

Table-1: Data set Description

Property	**Description**
Dataset Name	India 2025 Hourly Weather Dataset
File Format	CSV (Comma-Separated Values)
Total Records	8,760
Total Features	5
Data Type	Time-Series Data
Observation Frequency	Hourly
Forecast Target	Temperature
Year Covered	2025

2.3 Feature Engineering

Feature engineering is an important step in this project because it converts raw weather data into a suitable format for machine learning models because hourly weather data used in this project can not be directly used for daily forecasting.

The first step is date-time conversion where the date column is converted into proper date-time format. After that it is set as the index of the dataset. In the second step all the hourly temperature values are converted into daily average temperature. This step is known as daily sampling.

After sampling, some missing values are removed to make the dataset clean. This is called missing value handling. The fourth step is train-test splitting, where the daily temperature data is divided into training and testing parts. For the LSTM model the temperature values are scaled between 0 to 1 using MinMaxScaler. Thus, these overall feature engineering steps make the dataset clean, structured, and suitable for AI/ML models.

2.4 Machine Learning Models

It is the most important part of the whole project where different time-series forecasting models are used to predict future temperature values. As the weather data is time dependent time-series forecasting models are suitable for the weather prediction. In this project, three models are used for forecasting:

2.4.1 ARIMA Model

ARIMA stands for AutoRegressive Integrated Moving Average. It is a commonly used statistical model useful when the data contains a pattern or trend over time in time-series forecasting. It has three main parts:

A. AR- AutoRegressive Part: It means that the future value is predicted using previous values. The model studies the last few days temperature values and predicts the future values.

B.I- Integrated Part: It is used to make the data more stable. If the data shows increasing or decreasing trend over time ARIMA used differencing to remove all those trends and makes it suitable for forecasting.

C. MA-Moving Average part: The moving average part considers the previous forecasting errors and corrects itself by learning from past prediction mistakes. This improves the quality of the future predictions. The general equation of an ARIMA model is:

$$ARIMA(p, d, q)$$

Where: p = number of past values used, d = number of differencing operations , q = number of past error terms used In the given project code, the ARIMA model is created using the order:

2.4.2 SARIMA Model

SARIMA stands for Seasonal AutoRegressive Integrated Moving Average. SARIMA contains both non-seasonal and seasonal parameters. The non-seasonal part works like ARIMA, while the seasonal part captures repeating patterns in the time-series data. The general equation for the SARIMA model is:

$$SARIMA(p, d, q)(P, D, Q)s$$

where:

- p=autoregressive order
- d=differencing order
- q=moving average order
- P=seasonal autoregressive order
- D=seasonal differencing order
- Q=seasonal moving average order
- S=seasonal period

2.4.3 LTSM Model

LSTM stands for Long Short-Term Memory. It is a deep learning model used for sequence prediction problems. It is a special type of Recurrent Neural Network (RNN) which can learn patterns from previous data and remember important information for longer time. Before using LSTM, the data is scaled using MinMaxScaler into range between 0 and 1. Unlike traditional models, LSTM can capture complex and non-linear patterns in the dataset. As temperature may depend upon previous temperature LSTM learns these relations from historical data. After scaling, the data is converted into sequences. In this project, the sequence length is 10 days. The LSTM architecture used in this project contains:

- First LSTM layer with 50 units
- Second LSTM layer with 50 units
- Dense output layer with 1 neuron

The first LSTM layer uses return_sequences=True, which means it passes sequence output to the next LSTM layer. The second LSTM layer processes this information and sends the final output to the Dense layer. The Dense layer gives the final predicted temperature value.

The model is compiled using:

- Optimizer: Adam
- Loss function: Mean Squared Error

The Adam optimizer helps the model update its internal weights during training. The model is trained for 20 epochs with a batch size of 32. An epoch means one complete training cycle through the dataset. Batch size means the number of samples processed before updating the model weights.

2.5 Model Training

After data processing and feature engineering hourly data set is converted into daily average temperature data. 80% data has been used for training and 20% for testing. ARIMA and SARIMA are trained directly on the daily temperature data. LSTM requires additional preprocessing, such as scaling and sequence creation. After training, all three models generate predictions.

2.6 Model Evaluation using RMSE

The evaluation metric RMSE stands for Root Mean Squared Error. In this step RMS calculates the difference between actual temperature values.

3. RESULT AND IMPLEMENTATION

The implementation of the project includes two parts. The first part is the machine learning implementation, where ARIMA, SARIMA, and LSTM models are trained and tested. The second part is the application implementation, where the saved results are displayed using a Streamlit web application.

Table 2: Performance metrics

Model	RMSE Value	Performance Analysis
ARIMA	3.1858381427	Suitable for trend-based forecasting
SARIMA	2.3163921426	Suitable for seasonal pattern forecasting
LSTM	0.6074101536	Suitable for long-term sequence learning

Table 3: Temperature forecasting comparison among all models

Date	ARIMA Forecast	SARIMA Forecast	LSTM Forecast
2026-01-01 00:00:00	23.2762	22.8865	19.3321
2026-01-02 00:00:00	23.3138	22.8307	19.3465
2026-01-03 00:00:00	23.2964	22.6838	19.3768
2026-01-04 00:00:00	23.2014	22.5488	19.4028
2026-01-05 00:00:00	23.2250	22.6651	19.4368
2026-01-06 00:00:00	23.2184	22.4612	19.4614
2026-01-07 00:00:00	23.2053	22.3711	19.4781
2026-01-08 00:00:00	23.1815	21.9970	19.5119
2026-01-09 00:00:00	23.1736	21.9487	19.5602
2026-01-10 00:00:00	23.1753	21.7936	19.6037

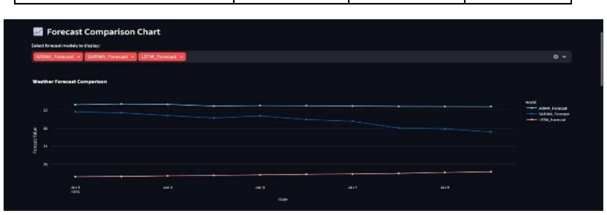

Figure 3: Visualization of performance metrics among all models

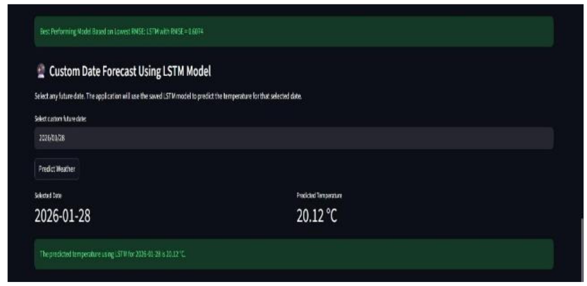

Figure 4: Temperature Forecasting using Predictor

4. CONCLUSION & FUTURE WORK

The Smart Weather Forecasting System developed in this project provides a simple and effective method for forecasting future temperature using Artificial Intelligence and Machine Learning techniques. The major conclusions of this project are:

a) Historical data supports future temperature prediction.

b) Time-series models are suitable for weather forecasting.

c) ARIMA, SARIMA, and LSTM offer different forecasting approaches.

d) RMSE identifies the most accurate model.

e) Graphs and CSV outputs simplify result analysis.

The system uses graphs to clearly visualize original data, forecasts, and RMSE comparisons. Future improvements can enhance the system's accuracy, practicality, and real-world usability.

REFERENCES

[1] BRAC University, (2022). Weather Forecasting using Deep Learning. BRAC University Thesis/Project Report, Department of Computer Science and Engineering.

[2] Waqas, M., (2025). Artificial Intelligence and Numerical Weather Prediction. ScienceDirect.

[3] Karevan, Z., & Suykens, J. A. K., (2018). Spatio-temporal Stacked LSTM for Temperature Prediction in Weather Forecasting. arXiv preprint arXiv:1811.06341. doi: 10.48550/arXiv.1811.06341.

[4] International Journal of Advanced Research in Science, Communication and Technology, (2025). Article on machine learning-based weather forecasting. International Journal of Advanced Research in Science, Communication and Technology (IJARSCT).

[5] Hossain, M. L., Shams, S. M. N., & Ullah, S. M., (2025). Time-series and deep learning approaches for renewable energy forecasting in Dhaka: A comparative study of ARIMA, SARIMA, and LSTM models. Discover Sustainability, 6, 775. doi: 10.1007/s43621-025-01733-5.

[6] Rozas Larraondo, P., (2019). Application of Machine Learning Techniques to Weather Forecasting. Doctoral Thesis, University of the Basque Country UPV/EHU, Department of Computer Science and Artificial Intelligence.

[7] Weather Forecasting using Machine Learning Techniques, (2020). Research/technical document on machine learning-based weather forecasting.

[8] Weather Forecasting using Machine Learning, (2026). Google Colab implementation/project. Google Colaboratory.

[9] Sanders, W. S., (2017). Machine Learning Techniques for Weather Forecasting. M.S. Thesis, University of Georgia, 136 pages.

[10] Karevan, Z., & Suykens, J. A. K., (2018). Spatio-temporal Stacked LSTM for Temperature Prediction in Weather Forecasting. arXiv preprint arXiv:1811.06341. doi: 10.48550/arXiv.1811.06341.