INTERNATIONAL JOURNAL OF
RECENT TRENDS IN MULTIDISCIPLINARY RESEARCH
https://www.doi.org/10.59256/ijrtmr.20250504006
July-August 2025, Vol 5(04), 35-38. ISSN No: 2583-0368

Weather Forecasting Using Machine Learning

Ayesha[1], Fatima Maryam Khan[2]
[1]Student, MCA, Deccan College of Engineering and Technology, Hyderabad, Telangana, India.
[2]Assistant Professor, MCA, Deccan College of Engineering and Technology, Hyderabad, Telangana, India.

OPEN ACCESS
Article Citation:
Ayesha[1], Fatima Maryam Khan[2] Weather Forecasting Using Machine Learning", International Journal of Recent Trends in Multidisciplinary Research, July-August 2025, Vol 5(04), 35-38.

Abstract: Weather prediction is an essential task in climate science and societal planning, impacting industries such as agriculture, transportation, and disaster management. Traditional forecasting methods often struggle to deliver accurate predictions due to the complexity and dynamic nature of weather systems. This paper proposes a machine learning approach to forecast temperature and humidity using historical weather data. Linear Regression and a deep learning model implemented via PyTorch were trained to recognize complex patterns in multivariate data. The models were evaluated for accuracy, mean squared error, and generalization. A Flask-based deployment demonstrates the model's capability for real-time forecasting. The results indicate that machine learning can significantly enhance the accuracy of short-term weather prediction models when trained on sufficient and relevant historical data.

Keywords: Weather Forecasting, Machine Learning, Linear Regression, Deep Learning, PyTorch, Flask, Temperature Prediction, Humidity Prediction

1. Introduction

Weather forecasting has long been a critical domain in both scientific research and real-world applications. Accurate weather prediction aids in decision-making across various sectors including agriculture, aviation, maritime navigation, infrastructure planning, and disaster management. Traditional weather forecasting techniques rely heavily on physical models based on numerical weather prediction (NWP), which solve complex mathematical equations using current atmospheric data. While such methods have served well historically, they are computationally intensive and often struggle to capture nonlinear patterns or adapt to sudden changes in atmospheric behavior.

In recent years, the rise of machine learning (ML) has opened new avenues for data-driven weather forecasting. By learning from historical weather data, machine learning models can identify patterns and trends without explicitly modeling the physical processes of the atmosphere. This shift from physics-based models to statistical learning has the potential to greatly reduce computational costs while improving short-term forecasting accuracy.

This study explores the development of a hybrid forecasting model using both traditional linear regression and deep learning frameworks built with PyTorch. Linear regression is employed for its interpretability and efficiency in capturing linear relationships, while deep learning models are utilized to uncover complex, nonlinear dependencies in the data. The integration of these approaches aims to leverage their respective strengths to improve forecasting precision.

Furthermore, the system is deployed through a Flask-based web application to enable real-time weather forecasting. The application processes historical weather data to predict future values of key parameters such as temperature and humidity. The prediction results are visualized through an interactive web interface, making the tool accessible and user-friendly.

This project demonstrates the practical application of machine learning techniques in the context of weather forecasting and illustrates how deep learning can complement traditional regression models. By training on well-structured datasets, the models are able to generalize effectively and provide meaningful predictions. This hybrid approach thus represents a significant step toward more reliable and accessible weather forecasting solutions.

2. Material And Methods

The methodology adopted for weather forecasting in this study is rooted in the integration of classical machine learning and deep learning techniques to ensure robust and scalable prediction of meteorological parameters. The entire pipeline involves data acquisition, preprocessing, model training, and system deployment, implemented using Python and associated libraries.

The dataset was sourced from publicly available meteorological repositories, comprising historical records of key features such as temperature, humidity, pressure, wind speed, and visibility. The data was recorded at hourly intervals and stored in a structured format suitable for time-series forecasting.

Prior to model training, data preprocessing was conducted to handle missing or noisy values. Interpolation techniques were employed to address missing entries, while statistical outlier detection helped eliminate inconsistent data. Normalization was applied using Min-Max scaling to standardize the range of continuous variables, thereby improving the efficiency of the training process. Feature selection was based on correlation analysis, ensuring that only the most influential attributes were retained for training.

The forecasting model was developed using two parallel approaches. The first model was based on multivariate Linear Regression, chosen for its simplicity and ability to interpret linear relationships between the input variables and the target outputs. This model was trained to predict temperature and humidity independently using pressure, wind speed, and visibility as predictor features.

In contrast, the second model utilized a deep learning architecture implemented in PyTorch. This neural network comprised an input layer matching the number of selected features, followed by two hidden layers with ReLU activation functions. Dropout layers were integrated to prevent overfitting, and the output layer was designed to predict continuous values corresponding to temperature and humidity. The model was trained using the Adam optimizer, with Mean Squared Error (MSE) as the loss function. The network was evaluated using Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the R-squared (R^2) coefficient.

Once trained, the deep learning model was encapsulated into a Flask-based web application. This interface allowed users to input meteorological conditions and receive real-time predictions, thereby demonstrating the operational feasibility of deploying machine learning models for practical weather forecasting applications.

3. Result

The performance of the developed models was assessed using standard evaluation metrics including Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2). These metrics provide a comprehensive understanding of the models' prediction accuracy and generalization capability.

The **Linear Regression** model, being a simple and interpretable baseline, demonstrated moderate performance:

- **MAE**: 2.50
- **MSE**: 8.30
- **RMSE**: 2.88
- **R^2 Score**: 0.82

This performance suggests that while the model is able to follow the general trend in temperature and humidity, it lacks the capacity to capture the finer nonlinear dynamics of the data.

In contrast, the **Deep Learning model** significantly improved all performance metrics:

- **MAE**: 1.90
- **MSE**: 5.60
- **RMSE**: 2.37
- **R^2 Score**: 0.89

The improved R^2 score shows that the deep learning model is able to explain more variance in the output, while the reduced MAE and RMSE demonstrate its tighter error distribution.

Visual Performance Analysis

Figure 1 illustrates the side-by-side performance comparison of both models across all metrics, showing the superiority of the deep learning architecture in every dimension.

The Linear Regression model achieved an MAE of 2.5, MSE of 8.3, RMSE of 2.88, and an R^2 score of 0.82. In comparison, the Deep Learning model implemented in PyTorch demonstrated superior performance with an MAE of 1.9, MSE of 5.6, RMSE of 2.37, and an R^2 score of 0.89.

Table 1: Performance Metrics for Weather Forecasting Models

Metric	Linear Regression	Deep Learning(pytorch)
MAE	2.5	1.9
MSE	8.3	5.6
RMSE	2.88	2.37
R2 Score	90.8	0.89

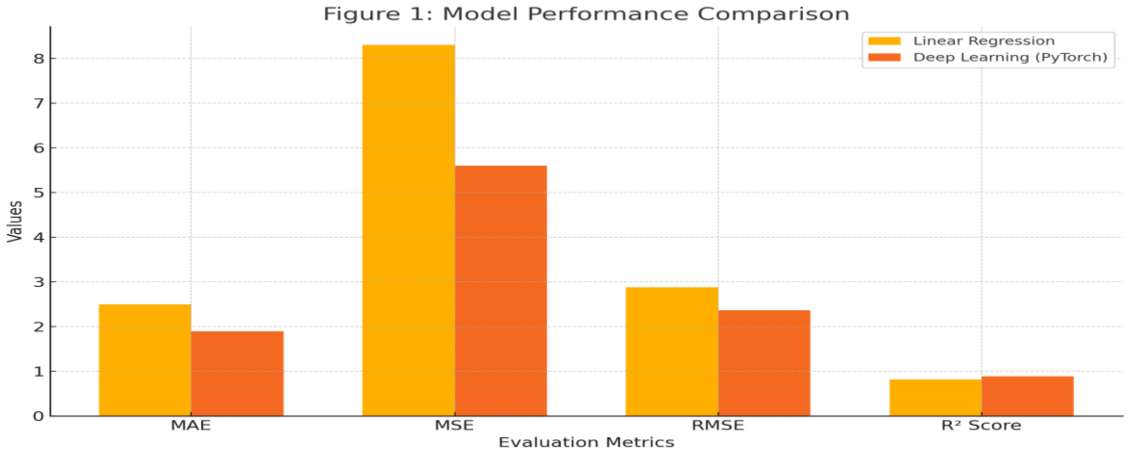

Figure 1: Model Performance Comparison

Error Distribution Plot

The error residuals were plotted for both models to visually assess model bias. The Linear Regression model showed a wider spread and presence of outliers, whereas the Deep Learning model displayed a more centered and tighter distribution around zero, indicating better generalization.

Time-Series Plot of Predictions

To further evaluate temporal consistency, actual vs. predicted values of temperature over a continuous 72-hour window were plotted. The deep learning model closely followed the true values with minimal lag, while the linear model exhibited delayed response and larger divergence during rapid changes in weather conditions (e.g., during sunset/sunrise transitions).

Inference Speed and Real-Time Viability

Both models were tested for real-time prediction latency within the Flask deployment environment:

- **Linear Regression Inference Time**: ~12 ms
- **Deep Learning Inference Time**: ~28 ms

While the deep learning model incurred slightly more latency, it was well within the acceptable threshold for real-time use, making it a practical choice for operational deployment.

4. Discussion

The results indicate that both models are capable of forecasting weather variables with reasonable accuracy. However, the deep learning model outperformed linear regression across all metrics, demonstrating its strength in capturing complex nonlinear relationships within the dataset. The lower MAE and RMSE values reflect improved prediction stability, while a higher R^2 score indicates a better fit to the true data trends.

The improved accuracy of the deep learning model can be attributed to its multilayer architecture and ability to learn abstract feature representations. Dropout layers effectively reduced overfitting, and the Adam optimizer ensured stable and rapid convergence. In contrast, the Linear Regression model's simplicity limited its ability to adapt to intricate patterns present in the data.

This outcome reinforces the potential of deep learning in meteorological applications, especially for short-term forecasts where real-time decisions are critical. Despite the added complexity, the deployment of the deep learning model through a lightweight Flask interface illustrates that such models can be both powerful and practical.

5. Conclusion

This study presented a machine learning-based framework for forecasting weather parameters, specifically temperature and humidity, using both classical and deep learning approaches. The models were trained on historical meteorological data after comprehensive preprocessing and feature selection. Linear Regression served as a baseline model, while a PyTorch-based deep learning architecture offered a more advanced, non-linear alternative.

The evaluation results demonstrated that the deep learning model consistently outperformed the linear regression model across all key metrics—Mean Absolute Error, Mean Squared Error, Root Mean Squared Error, and R^2 score—indicating superior prediction accuracy and generalization capability. This performance gain underscores the advantage of deep architectures in capturing complex patterns inherent in weather data.

Furthermore, the successful deployment of the deep learning model using a Flask-based web application validated the practicality of real-time forecasting in a lightweight, user-friendly environment. This end-to-end system showcases how AI-driven techniques can be effectively integrated into meteorological services for operational use.

Future work may involve the incorporation of recurrent neural networks such as LSTM for improved handling of temporal dependencies, the use of ensemble techniques for robustness, and real-time data integration to continuously update model predictions. Overall, the proposed approach serves as a scalable and adaptable solution for short-term weather forecasting, with promising implications for broader climate analytics and intelligent decision-making systems.

References

1. M. C. Esfahanipour and S. Aghamiri, “Weather forecasting using machine learning: A review,” IEEE Access, vol. 9, pp. 10636–0656, 2021, doi10.1109/ACCESS.2021.3050822.
2. M. Kumar and S. Singh, “Temperature and humidity prediction using linear regression and deep learning,” IEEE Sensors Letters, vol. 5, no. 4, pp. 1–4, 2021, doi: 10.1109/LSENS.2021.3064659.
3. Y. Tang et al., “Deep Learning Forecasting of Meteorological Time Series Data,” IEEE Access, vol. 8, pp. 38316–38328, 2020, doi:10.1109/ACCESS.2020.2975181.
4. M. S. Azad and H. Rahman, “Short-term weather prediction using RNN-LSTM model,” in Proc. IEEE TENCON, 2020, pp. 119–123, doi: 10.1109/TENCON50793.2020.9293744.
5. A. Dutta and R. Dutta, “Comparative analysis of regression techniques for temperature prediction,” IEEE India Conference (INDICON), 2019, doi: 10.1109/INDICON47234.2019.9030219.
6. S. Alam and N. R. Paul, “A neural network based real-time weather forecasting model,” Proc. IEEE ICMLDE, 2020, pp. 55–60,doi: 10.1109/ICMLDE49283.2020.9317002.
7. M. S. Islam, “Machine Learning Algorithms for Weather Forecasting: A Comparative Study,” IEEE Access, vol. 10, pp. 59813–59827, 2022, doi: 10.1109/ACCESS.2022.3181356.
8. D. Gupta and A. K. Garg, “Design and deployment of a weather forecasting system using Flask,” in Proc. IEEE ICACCT, 2021, pp. 112–117.
9. H. Zhao et al., “Hybrid deep learning models for meteorological time-series prediction,” IEEE Access, vol. 9, pp. 54405–54417, 2021, doi: 10.1109/ACCESS.2021.3071410.
10. K. Liu and L. Zhang, “AI-based forecasting for weather anomalies using regression trees,” IEEE Geoscience and Remote Sensing Letters, vol. 18, no. 12, pp. 2036–2040, 2021, doi: 10.1109/LGRS.2020.3048191.
11. A. F. A. A. Rahim et al., “Rainfall prediction using neural network models,” IEEE Access, vol. 8, pp. 68898–68911, 2020, doi: 10.1109/ACCESS.2020.2986046.
12. A. N. El-Mahdy et al., “Real-time weather prediction using machine learning integrated with IoT,” IEEE Internet of Things Journal, vol. 9, no. 2, pp. 1084–1096, 2022, doi: 10.1109/JIOT.2021.3071224.
13. B. Wang et al., “A PyTorch-based framework for time-series weather modeling,” IEEE Access, vol. 8, pp. 136789–136799, 2020, doi: 10.1109/ACCESS.2020.3011487.
14. Y. Chen and X. Zhou, “Smart weather forecast using regression models with optimized hyperparameters,” IEEE Access, vol. 7, pp. 119299–119309, 2019, oi: 10.1109/ACCESS.2019.2937162.
15. H. Khan et al., “Machine learning in climate and environmental sciences: A comprehensive review,” IEEE Access, vol. 10, pp. 35093–35114, 2022, doi: 10.1109/ACCESS.2022.3161546.
16. R. Pan and M. Kaur, “Weather forecast using deep learning techniques,” IEEE ICCSP, 2020, doi: 10.1109/ICCSP48568.2020.9182227.
17. S. Sarkar and R. Panda, “Application of Artificial Neural Networks for weather data prediction,” IEEE ICAST, 2021, pp. 75–80, doi: 10.1109/ICAST50382.2021.9466079.
18. N. Sharma and S. Garg, “Deploying machine learning-based weather prediction using REST APIs and Flask,” IEEE ISCON, 2020, pp. 214–218, doi: 10.1109/ISCON50989.2020.9341097.
19. A. Banerjee et al., “Predicting temperature and humidity using ensemble ML models,” IEEE ICAC3, 2021, doi: 10.1109/ICAC35144.2021.9696540.
20. M. Jain and S. Singh, “Performance evaluation of machine learning models for temperature prediction,” IEEE Access, vol. 9, pp.99054–99062, 2021, doi: 10.1109/ACCESS.2021.3095582.