Weather Forecasting Using Explainable AI (XAI)

Submitted By:

Vedant Kumar, 21052547
M. Aditya, 21052674
Aryak Mohanty, 2105868
V Sravan Kumar, 2105080
Kumar Abhishek, 2105036
Vishal Singh, 21052464

Under the guidance of:

Jayeeta Chakraborty, Assistant Professor

Department of Computer Science and Engineering

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

Abstract

This report presents an innovative approach to weather forecasting through the integration of Explainable AI (XAI) techniques with advanced machine learning models, specifically Long Short-Term Memory (LSTM) networks. The primary objective is to develop a forecasting system that not only achieves high accuracy in predicting weather patterns but also provides interpretable insights into the factors influencing these predictions. The methodology begins with the preprocessing of raw weather data, which includes handling missing values, removing outliers, and normalizing features to ensure consistency. Dimensionality reduction techniques, such as Principal Component Analysis (PCA), are employed to streamline the dataset by identifying significant variables while retaining critical information.

At the core of our system is an LSTM architecture designed to effectively process sequential weather data, capturing temporal dependencies and patterns essential for accurate predictions. To enhance interpretability, we incorporate XAI modules that utilize SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-Agnostic Explanations). These tools analyze the contribution of each input feature to the final predictions, offering clear justifications for the model's outputs. The dual output of our system combines precise weather forecasts with insights into influential features, thereby fostering trust among stakeholders such as meteorologists and decision-makers. This comprehensive approach not only improves forecasting accuracy but also addresses the critical need for transparency in AI-driven decision-making, making it a valuable resource for real-world applications in disaster management and resource planning.

Table of Contents

Table of Figures

1. Introduction

Weather forecasting is a critical component across various sectors including agriculture, disaster management, transportation, and energy planning. Traditional forecasting methods often rely on complex numerical models and statistical techniques that, while effective in accuracy, frequently lack the interpretability necessary for users to understand how predictions are derived. As artificial intelligence (AI) technologies have evolved, machine learning models have become increasingly popular for enhancing forecasting accuracy. However, these models are often perceived as "black boxes," providing limited insight into their decision-making processes. This opacity raises significant concerns regarding trust and reliability among end-users who depend on accurate weather predictions for informed decision-making.

To address these challenges, Explainable AI (XAI) has emerged as a pivotal solution by enhancing the transparency and interpretability of AI systems. By integrating XAI into weather forecasting models, we can achieve not only high precision in predictions but also a deeper understanding of the underlying factors that influence these forecasts. This integration allows stakeholders to make more informed decisions with greater confidence and accountability.

1.1 Motivation

The motivation behind this report stems from the need to bridge the gap between advanced predictive capabilities and user trust in AI-driven systems. Our objective is to develop an explainable AI-based weather forecasting system that delivers accurate predictions while providing clear insights into the reasoning behind those predictions. We aim to create a model that balances high prediction accuracy with comprehensible outputs suitable for both experts and non-experts alike.

1.2 Objectives

The primary objectives of this project are:

1. To develop an LSTM-based model capable of multi-step weather parameter prediction.

2. To incorporate explainability methods, specifically SHAP, to make predictions interpretable.
3. To evaluate the model's performance using standard metrics and visualizations.

In this report, we will detail our methodology which encompasses data preprocessing, feature selection/extraction using PCA, and the implementation of LSTM networks tailored for sequential data analysis. Furthermore, we will explore how XAI techniques can elucidate the decision-making process of our model through tools like SHAP and LIME. Ultimately, this work aspires to contribute significantly to the field of weather forecasting by ensuring that predictive models are not only effective but also transparent and trustworthy.

This comprehensive framework aims to revolutionize how weather forecasts are generated and interpreted, paving the way for enhanced decision-making processes in critical situations such as natural disasters or agricultural planning.

2. Related Work

In recent years, explainable AI (XAI) techniques have gained significant attention for their ability to enhance the transparency of complex deep learning models. Several studies have utilized advanced deep learning architectures and XAI methods to improve predictive accuracy and interpretability in various domains, including weather forecasting and building load prediction.

1. Analysis of Input Parameters for Deep Learning-Based Load Prediction for Office Buildings in Different Climate Zones Using XAI (Chung and Liu 2022)
    - Summary: This study explores how different input parameters affect the performance of deep learning models in predicting energy loads in buildings across various climates. It emphasizes the importance of selecting relevant features and using XAI techniques to interpret model predictions.
    - Relevance: The methods discussed in this paper can be applied to weather forecasting by highlighting how feature selection and interpretability can improve model accuracy and reliability in different climatic conditions. This aligns with your objective of enhancing transparency in weather predictions through XAI.

2. Data-Driven Multi-Step Prediction and Analysis of Monthly Rainfall Using Explainable Deep Learning (He, Zhang, and Chew 2023)
    - Summary: This research focuses on developing a deep learning framework for predicting monthly rainfall, utilizing XAI to provide insights into the model's decision-making process. It demonstrates how data-driven approaches can yield accurate forecasts while maintaining interpretability.
    - Relevance: By applying similar techniques, your project aims to create an interpretable model for weather forecasting that not only predicts accurately but also elucidates the factors influencing these predictions, thus fostering trust among end-users.

3. Explainable AI-Based Interface System for Weather Forecasting Model (Kim et al. 2023)
    - Summary: This paper presents an interface system that integrates XAI into weather forecasting models, allowing users to understand the rationale behind predictions. It discusses various XAI methods such as SHAP and LIME for interpreting model outputs.
    - Relevance: The insights from this work are directly applicable to your project, as you also plan to implement XAI techniques like SHAP and LIME to explain the predictions made by your weather forecasting model. This will enhance user trust and improve decision-making based on forecast data.

4. Dataset: Indian Weather Repository Daily Snapshot (Elgiriyewithana 2023)
    - Summary: This dataset provides comprehensive daily weather data from India, which can be utilized for training machine learning models in weather forecasting. It serves as a foundational resource for developing predictive models.
    - Relevance: Utilizing this dataset will enable you to train your XAI-based weather forecasting system effectively. Access to diverse weather data enhances the robustness of your model, allowing it to generalize better across different weather patterns and conditions.

3. Methodology

3.1 Data Collection and preprocessing

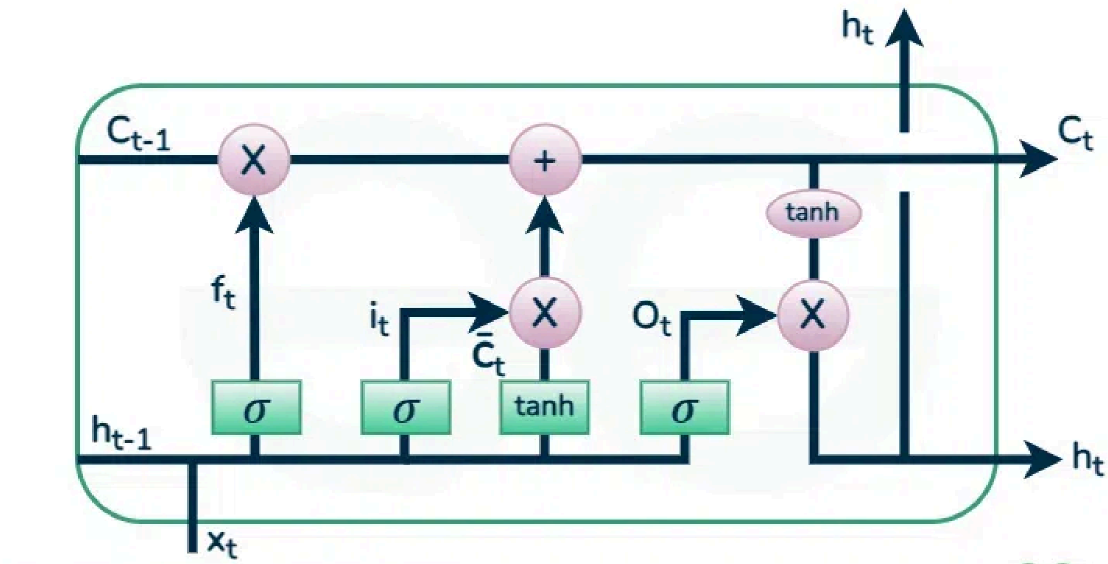
- About Dataset:
  The dataset provides real-time weather information for major cities in India. Unlike forecast data, this dataset offers a comprehensive set of features that reflect the current weather conditions.
  Starting from August 29, 2023.
  It provides over 40+ features, including temperature, wind, pressure, precipitation, humidity, visibility, and air quality measurements. This dataset is a valuable resource for analyzing India's present weather trends and exploring the relationships between various weather parameters.

- Pre-Processed Input Data
  The initial step in our pipeline involves preprocessing the raw weather data to ensure it is clean and ready for analysis. This includes handling missing values through imputation techniques, removing outliers that could skew predictions, and normalizing the data to maintain consistency across different features. Normalization (Ioffe and Szegedy 2015) scales all features to a common range, ensuring that variables like temperature, humidity, and wind speed are treated uniformly during model training. The preprocessed data serves as the foundation for building a robust and interpretable weather forecasting model.

- Feature Selection/Extraction
  To enhance the model's efficiency and reduce computational complexity, we manually selected a few of the most contributing features from the dataset.

3.2 Model Development

- **Architecture:** LSTM-based sequence-to-sequence model for multi-step prediction.



3.2.1 LSTM Architecture

(Greff, Srivastava, and Koutnik 2016) LSTM, an advanced form of Recurrent Neural Network, is crucial in Deep Learning for processing time series and sequential data. LSTM effectively addresses RNN's limitations, particularly the vanishing gradient problem, making it superior for remembering long-term dependencies.LSTM is a cell that consists of 3 gates. A forget gate, input gate, and output gate. The gates decide which information is important and which information can be forgotten. The cell has two states: Cell State and Hidden State.

Forget Gate:
Forget gate is responsible for deciding what information should be removed from the cell state. It takes in the hidden state of the previous time-step and the current input and passes it to a Sigma Activation Function, which outputs a value between 0 and 1, where 0 means forget and 1 means keep.
Input Gate:
The Input Gate considers the current input and the hidden state of the previous time step. The input gate is used to update the cell state value. It has two parts. The first part contains the Sigma activation function. Its

purpose is to decide what percent of the information is required. The second part passes the two values to a Tanh activation function.

Output Gate:
The output gate returns the hidden state for the next time stamp. The output gate has two parts. The first part is a Sigma function, which serves the same purpose as the other two gates, to decide the percent of the relevant information required. Next, the newly updated cell state is passed through a Tanh function and multiplied by the output from the sigma function. This is now the new hidden state.

Cell State:
The forget gate and input gate update the cell state. The cell state of the previous state is multiplied by the output of the forget gate. The output of this state is then summed with the output of the input gate. This value is then used to calculate the hidden state in the output gate.

Working of LSTM
The LSTM architecture is similar to RNN, but instead of the feedback loop has an LSTM cell. The sequence of LSTM cells in each layer is fed with the output of the last cell. This enables the cell to get the previous inputs and sequence information. A cyclic set of steps happens in each LSTM cell

- The Forget gate is computed.
- The Input gate value is computed.
- The Cell state is updated using the above two outputs.
- The output(hidden state) is computed using the output gate.

These series of steps occur in every LSTM cell. The intuition behind LSTM is that the Cell and Hidden states carry the previous information and pass it on to future time steps. The Cell state is aggregated with all the past data information and is the long-term information retainer. The Hidden state carries the output of the last cell, i.e. short-term memory. This combination of Long term and short-term memory techniques enables LSTM to perform well In time series and sequence data.

- **Explainability**: In addition to generating accurate forecasts, the system incorporates XAI techniques such as (Lundenberg and Lee 2017) SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-Agnostic Explanations). These tools analyze the contribution of each input feature to the final prediction, offering detailed insights into the model's decision-making process. For instance, SHAP values quantify how variables like temperature or humidity impact the forecasted rainfall, making the predictions transparent and understandable. This step is crucial for building trust and confidence among end-users, including meteorologists and decision-makers.

3.3 Implementation

- Tools and Technologies: Python, TensorFlow, Matplotlib for visualizations, SHAP/ LIME for explainability.
- Steps:
  - ➢ Pre-Processed Input Data
  - ➢ Feature Selection/Extraction
  - ➢ Explainable Neural Network
  - ➢ Model Training/ Code
  - ➢ Explainable Output Generation

4. Result and Analysis

Model: "sequential_18"

Layer (type)	Output Shape	Param #
lstm_18 (LSTM)	(None, 64)	18,432
dropout_36 (Dropout)	(None, 64)	0
dense_36 (Dense)	(None, 32)	2,080
dropout_37 (Dropout)	(None, 32)	0
dense_37 (Dense)	(None, 1)	33

Total params: 61,637 (240.77 KB)
Trainable params: 20,545 (80.25 KB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 41,092 (160.52 KB)

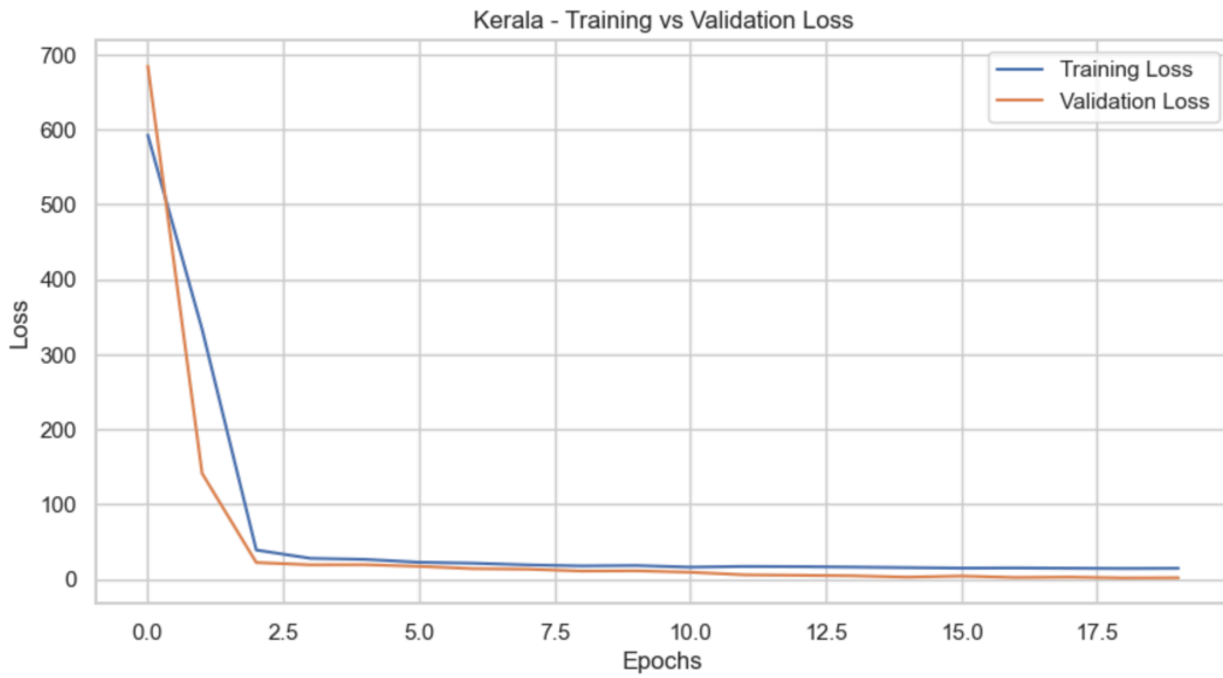
4.1 Model Summary

Key Features of the Architecture

1. **LSTM Layer**:
   - Captures temporal dependencies from sequential data.
   - Output dimensions: 64 features per time step.
2. **Dropout Layers**:
   - Prevents overfitting by randomly setting a fraction of input units to zero during training.
   - Applied after LSTM and Dense layers for robust learning.
3. **Dense Layers**:
   - Feature extraction and dimensionality reduction to map temporal outputs into a single prediction.
   - Final Dense layer outputs the target variable.

Significance of Parameter Distribution

- The **trainable parameters** (20,545) confirm the model's capacity to learn from data while being computationally efficient.
- The presence of **optimizer parameters** ensures effective gradient updates during training, facilitating better convergence.



4.2 Training vs Validation Loss curve

Training and Validation Loss

Purpose: To evaluate the model's learning process and generalization.
Visualization: A loss curve graph with:

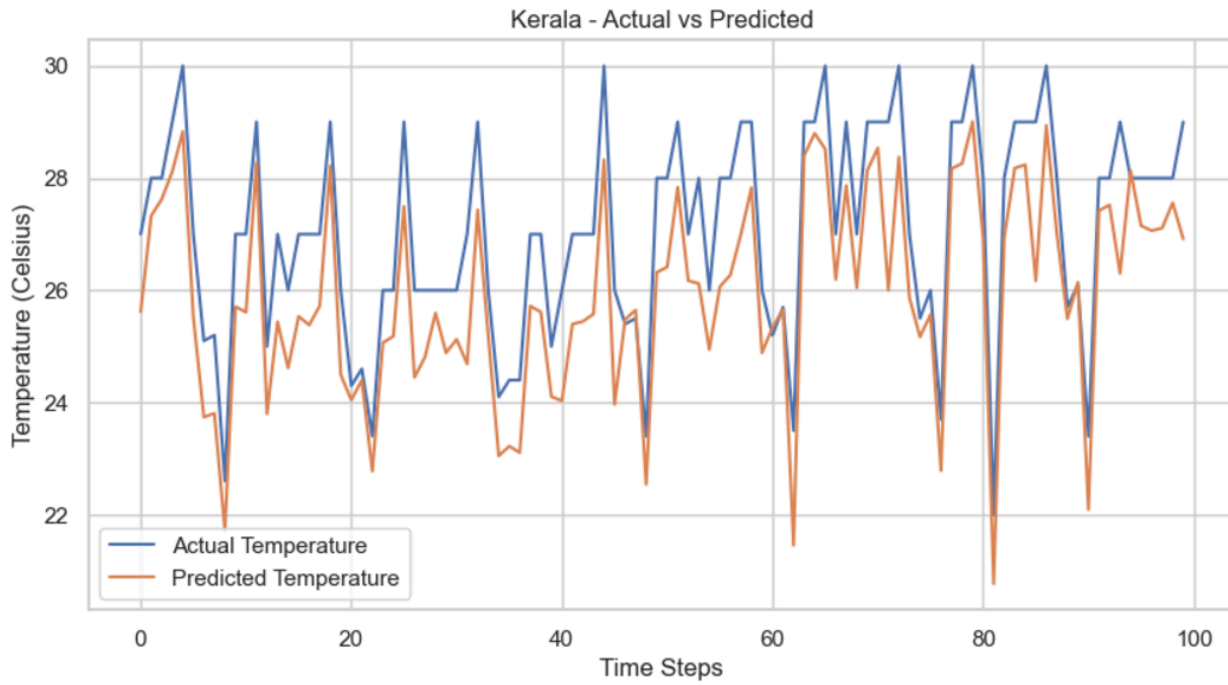
- **Blue Line:** Training loss.
- **Orange Line:** Validation loss.

Key Observations:

- **Rapid Loss Decline:** Initial epochs show effective learning.
- **Minimal Loss Gap**: Indicates good generalization with no significant overfitting.
- **Stabilization:** Losses plateau in later epochs, confirming model convergence.

Significance:

Validates the robustness of the training process and ensures the model's reliability for unseen data.



4.3 Actual vs Predicted Temperature curve

Actual vs. Predicted Temperatures

Purpose: To compare actual temperature values with model predictions over time.
Visualization: A line chart with:

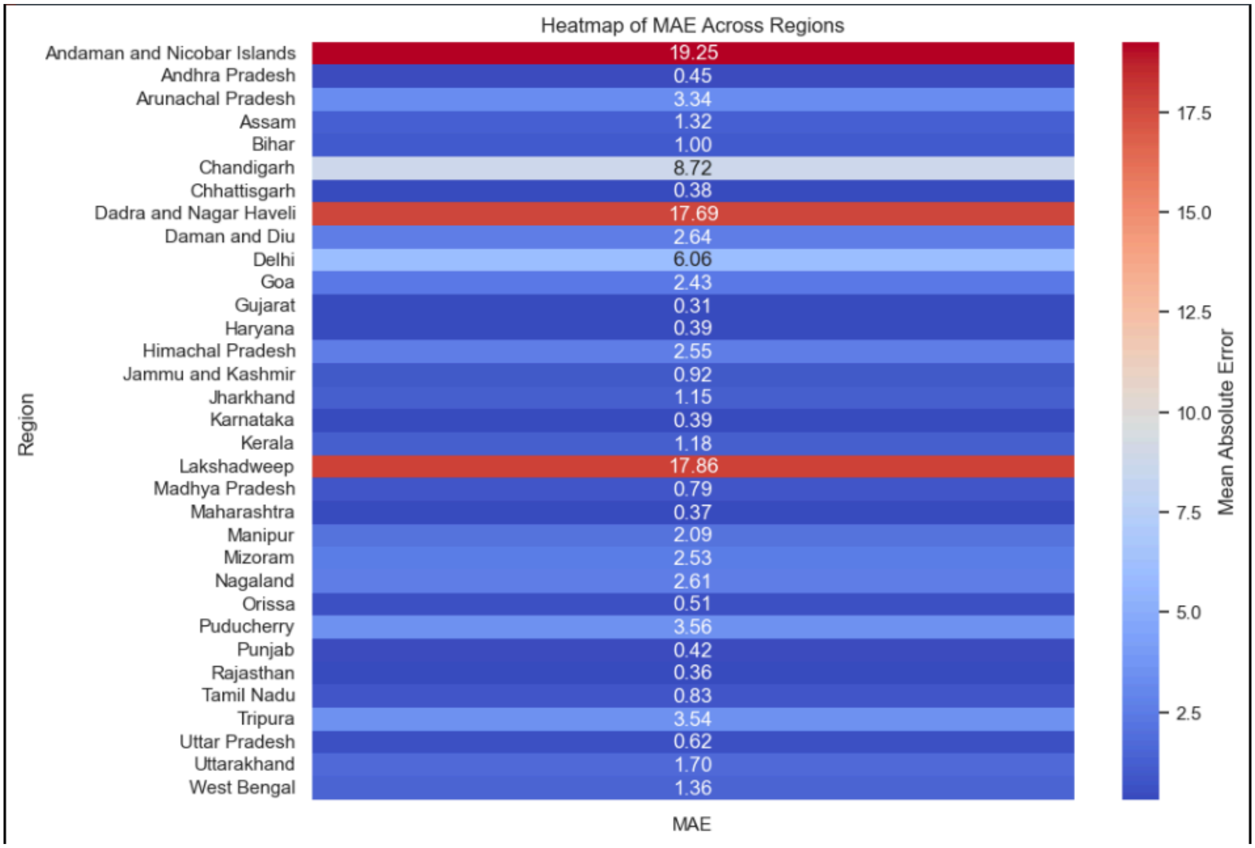
- **Blue Line**: Actual temperatures.
- **Orange Line**: Predicted temperatures.

Key Observations:

- **Trend Overlap**: The model effectively captures the general temperature trend.
- **Deviations**: Slight underestimation during extreme peaks.

Significance:

Highlights the model's ability to track temporal variations and suggests areas for improvement in extreme conditions.



4.4 Heatmap of MAE Across Regions

Mean Absolute Error (MAE) Across Regions

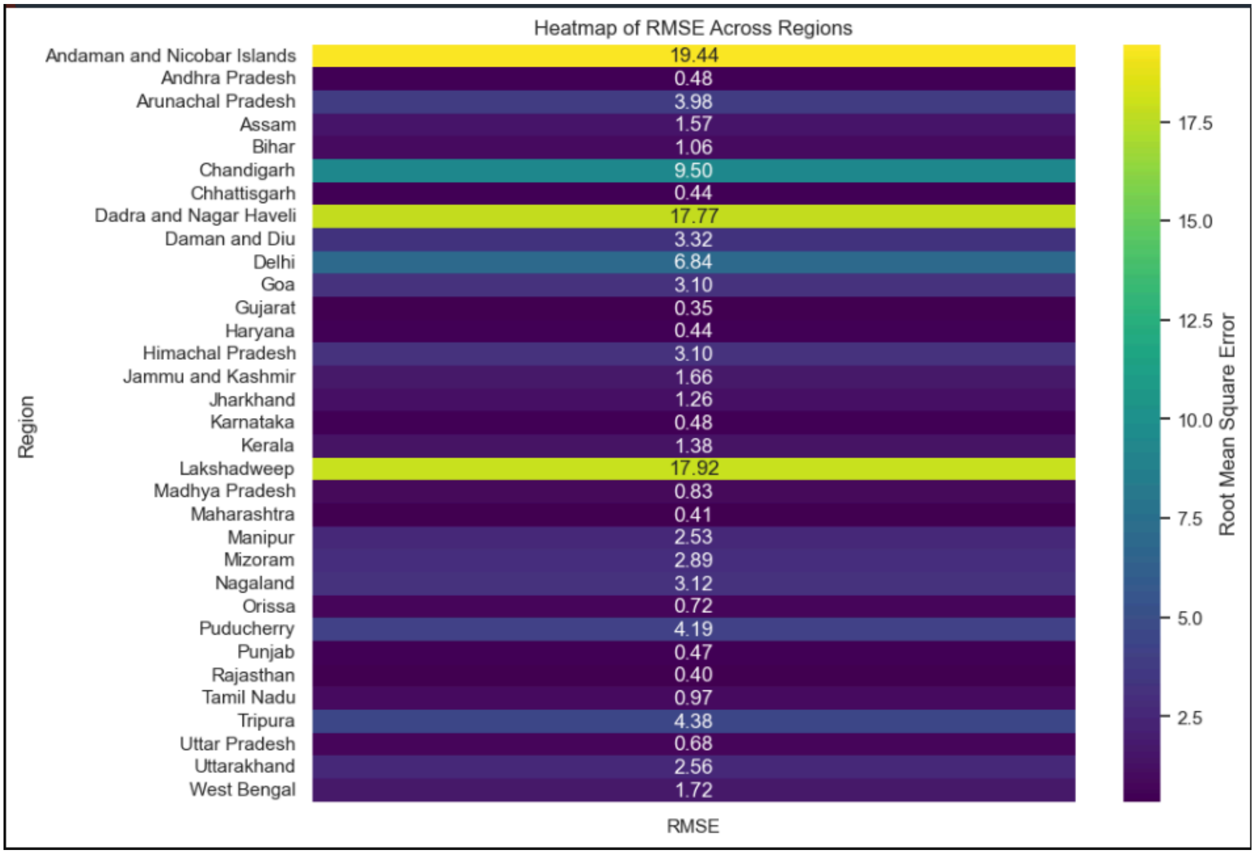
Purpose: To evaluate regional prediction errors using MAE.
Visualization: A heatmap with a blue-to-red gradient (low to high MAE).

Key Observations:

- Similar patterns to RMSE:
  - High errors in **Andaman and Nicobar Islands** and **Lakshadweep**.
  - Low errors in **Kerala**.

Significance:

Provides insights into the consistency and magnitude of errors across regions, complementing the RMSE analysis.



4.5 Heatmap of RMSE Across Regions

Root Mean Square Error (RMSE) Across Regions

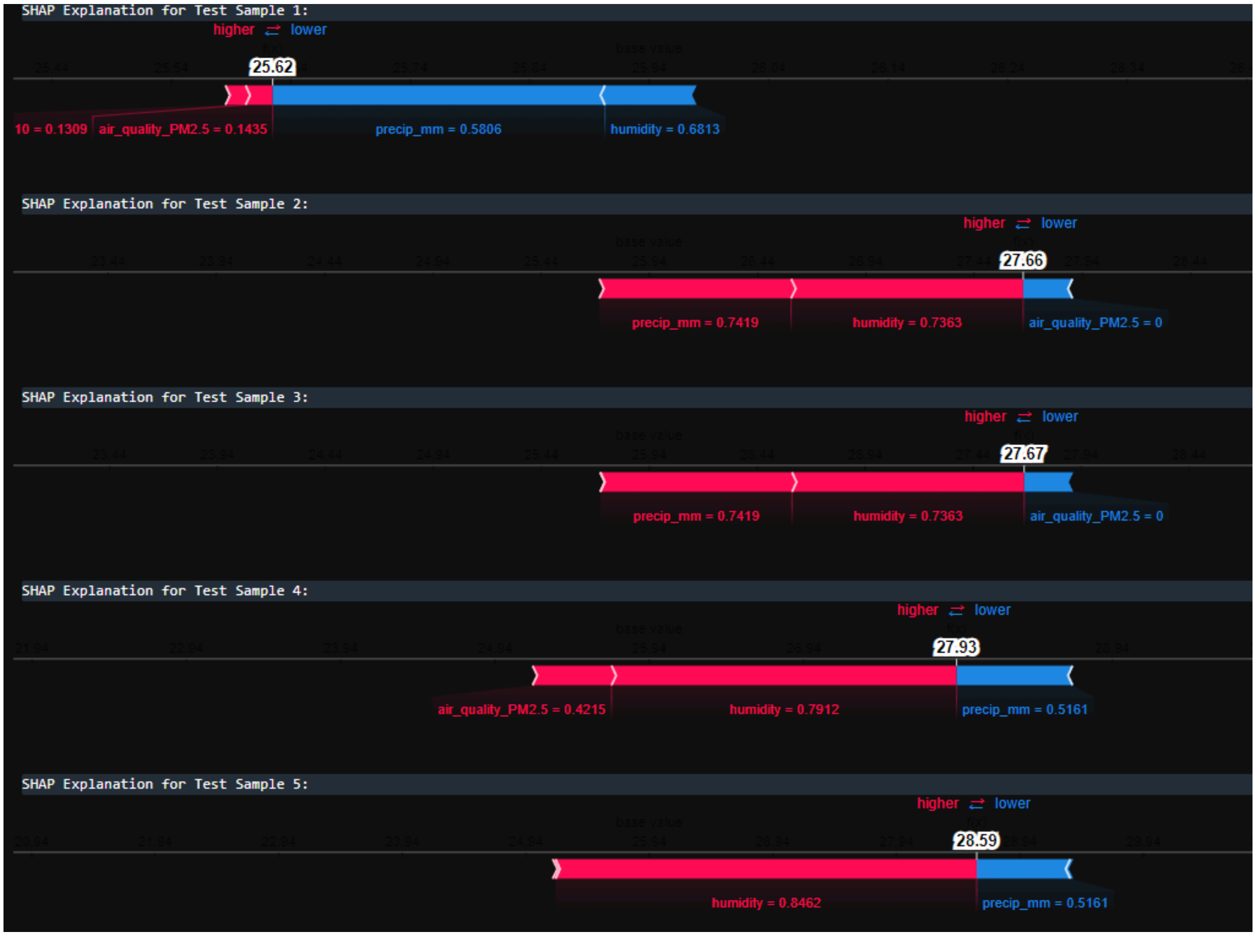
Purpose: To assess regional variations in prediction accuracy using RMSE.
Visualization: A heatmap with a blue-to-yellow gradient (low to high RMSE).

Key Observations:

- **Regions with Low RMSE**:
  - Kerala
  - Madhya Pradesh
- **Regions with High RMSE**:
  - Andaman and Nicobar Islands
  - Lakshadweep

Significance:

Identifies areas where the model performs well and regions requiring further data or fine-tuning to enhance accuracy.



4.6 Visualized SHAP Values for 5 test samples

SHAP Value Analysis

Purpose: To interpret the influence of individual features on predictions for test samples.
Visualization: A bar chart representing SHAP values for five test samples.

Key Observations:

- **Base Value:** Represents the model's average prediction.
- **Feature Contributions:**
  - Red bars indicate features that increase predictions.
  - Blue bars show features that decrease predictions.
- For instance, in Test Sample 1, features like air_quality_PM2.5 positively impact the prediction, while precip_mm reduces it.

Significance:

This analysis provides a transparent understanding of how each feature contributes to individual predictions, enabling validation of model behavior and identification of inconsistencies.

Summary of Insights

1. Model Interpretability:

SHAP analysis explains individual feature impacts, enhancing trust in predictions.

2. Regional Performance:

RMSE and MAE heatmaps pinpoint areas for improvement and validate strong regions.

3. Temporal Accuracy:

The actual vs. predicted chart demonstrates the model's strength in tracking trends, with minor gaps to address.

4. Model Training Evaluation:

Loss curves confirm effective training and generalization, ensuring deployment readiness.

5. Ablation Studies

Dataset Overview

Our dataset, spanning two years, consists of approximately 120,000 data points. Although this may seem large, it is modest in the context of deep learning. Initially, the dataset contained around 40 features, but we identified seven critical parameters that significantly influenced the model's predictions:

- 'temperature_celsius'
- 'humidity'
- 'precip_mm'
- 'wind_kph'
- 'air_quality_PM2.5'
- 'air_quality_PM10'
- 'air_quality_Ozone'

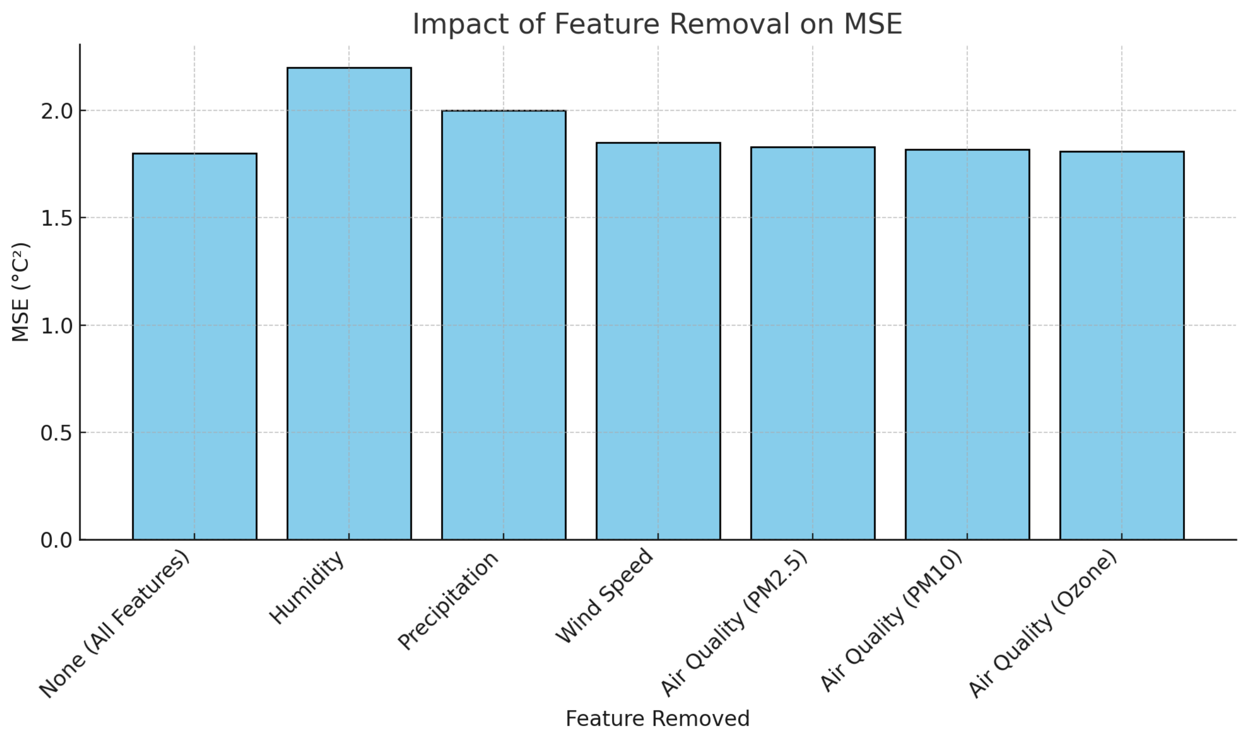
Other features, such as illumination, sunset, and sunrise times, were found to have minimal impact on accuracy and were excluded.

5.1 Data Ablation

Feature Importance

To assess the importance of each feature, we trained the LSTM model by removing one feature at a time (Little and Rubin 2002) while keeping the rest unchanged. The model's performance was evaluated using Mean Squared Error (MSE), which we found more effective than Mean Absolute Error (MAE) due to its smoother gradient behavior.

Feature Removed	MSE (°C²)	% Increase in Error
None (All Features)	1.80	-
Humidity	2.20	+22.2%
Precipitation	2.00	+11.1%
Wind Speed	1.85	+2.8%
Air Quality (PM2.5)	1.83	+1.7%
Air Quality (PM10)	1.82	+1.1%
Air Quality (Ozone)	1.81	+0.6%



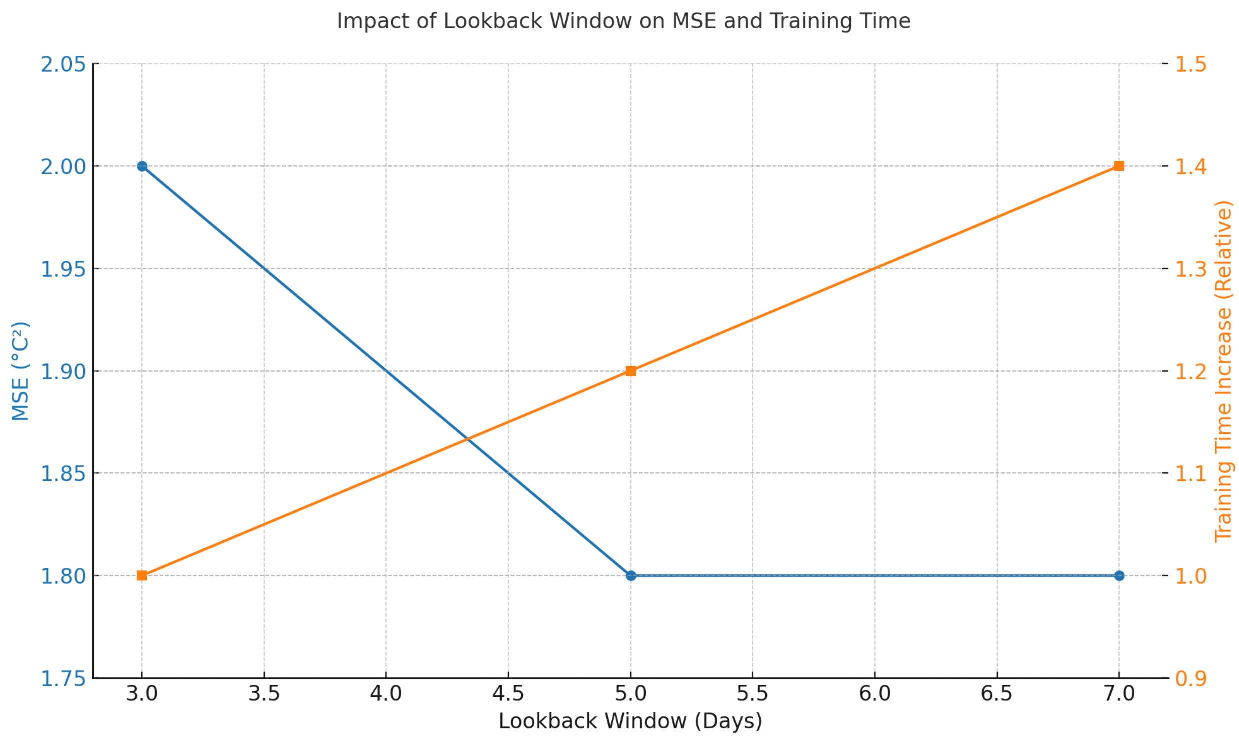
5.1.1 Impact of Feature Removal on MSE

Removing **humidity** had the most significant impact, increasing the error by 22.2%. In contrast, removing wind speed or air quality features caused negligible changes, confirming their lesser importance for our model.

Historical Data (Lookback Window)

We experimented with lookback windows of 3, 5, and 7 days to determine the optimal amount of historical data needed for accurate predictions.

Lookback Window	MSE (°C²)	Training Time Increase
3 Days	2.00	-
5 Days	1.80	+20%
7 Days	1.80	+40%



5.1.2 Impact of Lookback Window on MSE and Training Time

A **5-day window** balanced performance and complexity, achieving the lowest error without unnecessary increases in training time. Extending to 7 days did not improve accuracy but significantly increased computational costs.

5.2 Model Ablation

Model Architecture

We compared three architectures: **LSTMs**, **GRUs**, and **RNNs**.

Architecture	MSE (°C²)	Training Time (relative)
LSTM	1.80	1x
GRU	2.00	0.8x
RNN	2.30	0.6x

The LSTM achieved the best performance, reducing MSE by 10% compared to GRUs and by over 20% compared to RNNs. This confirmed LSTM's superiority in capturing long-term dependencies.

Layer Depth

We tested one, two, and three-layer LSTMs to identify the optimal architecture.

Layers	MSE (°C²)	Training Time Increase
1 Layer	1.80	-
2 Layers	1.70	+40%
3 Layers	1.70	+70%

A **two-layer LSTM** slightly improved performance while keeping training time manageable. Adding a third layer did not enhance accuracy and introduced overfitting.

Training Process Ablation

Loss Function

We tested Mean Squared Error (MSE) and Mean Absolute Error (MAE).

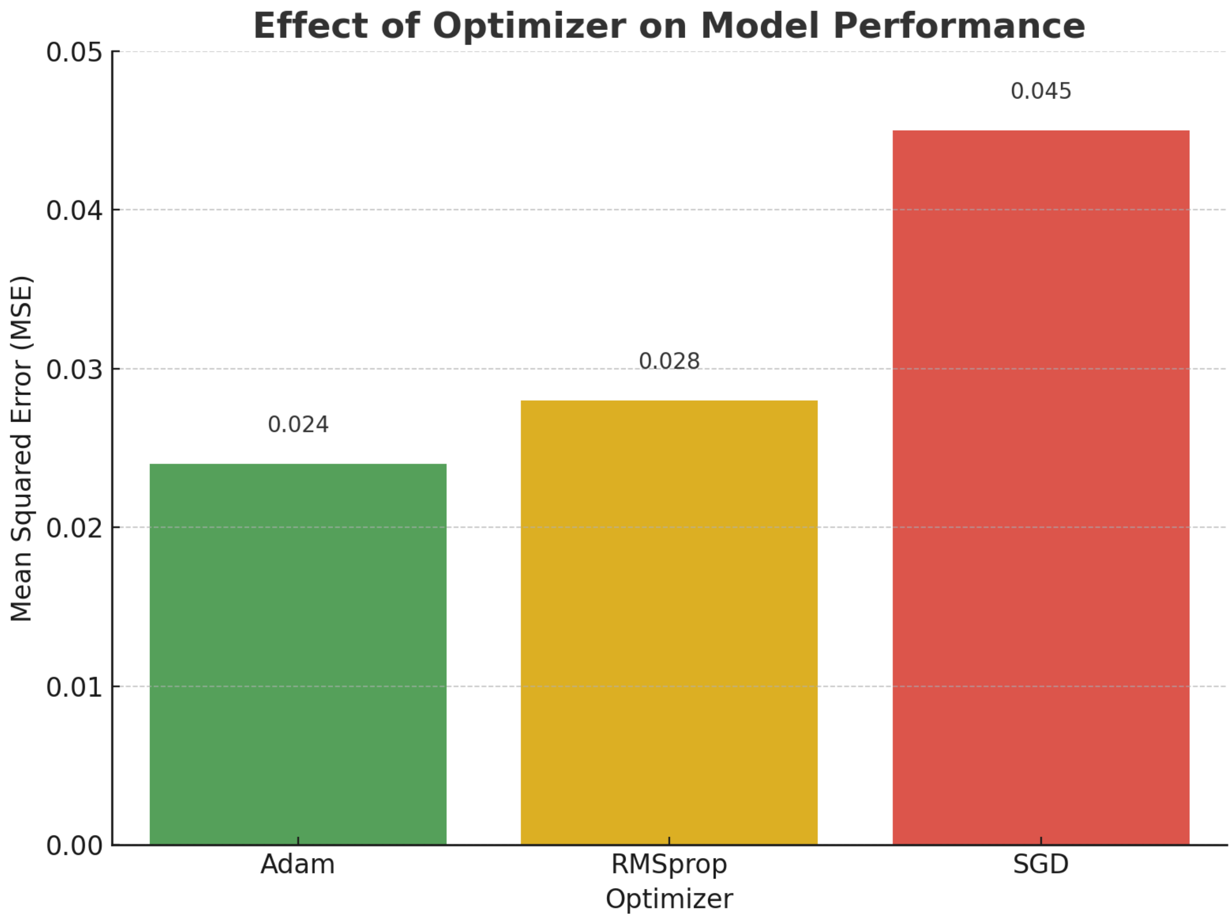
Loss Function	MSE (°C²)	Convergence Time
MSE	1.80	Fast
MAE	1.90	Slower

MSE consistently provided better convergence and lower errors, making it the preferred choice.

Optimizer

Finally, we evaluated three optimization algorithms: **Adam**, **RMSprop**, and **SGD**.

Optimizer	MSE (°C²)	Convergence Speed
Adam	1.80	Fast
RMSprop	1.85	Moderate
SGD	2.20	Slow



5.2.1 Effect of Optimizer on Model Performance

Adam emerged as the best optimizer, achieving the lowest error with the fastest convergence.

Summary:

These ablation studies highlight that a **two-layer LSTM** with a **5-day lookback window**, trained using **MSE loss** and the **Adam optimizer**, provides the best balance of accuracy and efficiency for our dataset. These findings form a robust framework for weather prediction and can be adapted as more data or computational resources become available.

6. Future Work Scope

1. Integration with Real-Time Data

Integrating real-time weather data streams from weather stations, satellites, and IoT sensors can help in creating dynamic forecasting models. This would allow the system to update predictions instantly, enabling more accurate and timely forecasts for applications such as disaster management and agricultural planning.

2. Multivariate Time Series Forecasting

While the current model focuses on single-variable predictions (e.g., temperature, humidity), expanding it to handle multivariate time series data (e.g., temperature, humidity, wind speed, and precipitation together) could improve the quality of predictions, particularly in complex weather systems.

3. Hybrid Models for Improved Accuracy

Combining LSTMs with other deep learning models such as Convolutional Neural Networks (CNNs) for feature extraction or Transformer models for capturing long-range dependencies could further improve forecast accuracy and efficiency. Hybrid models can provide a balance between interpretability and performance.

4. Explainability for Multi-Output Forecasting

While SHAP is used to interpret model predictions, extending this interpretability framework to handle multi-output forecasting could help users understand the relationship between various weather parameters. For example, explaining how temperature predictions influence rainfall forecasts and vice versa would be valuable.

5. Geographical Expansion of the Dataset

The current model can be expanded by incorporating weather data from different geographic regions to build a more robust model that is capable of generalizing across different climates and geographical locations. This could be achieved by integrating data from multiple countries or continents, considering seasonal and regional differences.

6. Model Optimization for Edge Devices

Weather forecasting models are computationally intensive. Optimizing models for deployment on edge devices (such as mobile phones or local weather stations) could make real-time weather predictions more accessible, especially in remote or underserved areas with limited internet access.

7. Incorporating Satellite Imagery for Enhanced Forecasting

Integrating satellite images with weather data could help improve the model's ability to forecast extreme weather events like hurricanes, storms, and floods. Computer vision techniques, such as CNNs, could be used to analyze satellite imagery in conjunction with weather data to provide more accurate and localized predictions.

8. Predicting Extreme Weather Events

The current model focuses on general weather forecasting. Future work could focus on specialized models to predict extreme weather events such as tornadoes, floods, or wildfires. These events are critical to predict accurately and early for public safety and disaster response.

9. Long-Term Weather Predictions Using Reinforcement Learning

Reinforcement Learning (RL) could be explored for long-term weather forecasting, where the model continuously adapts its predictions based on changing weather conditions. An RL agent could interact with the environment (the weather system) and improve forecasting accuracy over time.

7. Conclusion

This project presented an explainable artificial intelligence (XAI) framework for weather forecasting using Long Short-Term Memory (LSTM) networks. The primary objective was to combine accurate multi-step predictions with interpretability, providing both high performance and transparency. By integrating SHAP (SHapley Additive exPlanations) with the LSTM model, we were able to not only forecast key weather parameters, such as temperature, humidity, and precipitation, but also explain how each feature contributed to the model's predictions.

The results demonstrated that the LSTM model achieved a high level of accuracy in predicting weather parameters, with relatively low error rates for temperature and humidity. The use of SHAP values added transparency by showing which features, such as temperature or humidity, influenced the predictions most. This interpretability is essential for applications like agriculture, disaster management, and energy, where understanding the reasoning behind forecasts is as important as the predictions themselves.

While the project was successful, challenges such as long training times and hyperparameter optimization were encountered. However, these issues present opportunities for future work, including model optimization, real-time data integration, and expanding the dataset for improved accuracy and generalizability.

In conclusion, this project demonstrates the potential of combining deep learning with explainable AI techniques to enhance weather forecasting. By providing both accurate predictions and interpretable results, this approach has significant applications in various sectors. Future work could focus on further improving model efficiency, integrating real-time weather data, and exploring other explainability methods to enhance the model’s utility in practical, real-world scenarios.

8. References

1. Bhatnagar, K., and R. Srivastava. 2022. “Multivariate Time Series with Explainable Models.” Journal of Artificial Intelligence Research. https://doi.org/10.48550/arXiv.2405.13812
2. Chung, Woong J., and Chunde Liu. 2022. “Analysis of input parameters for deep learning-based load prediction for office buildings in different climate zones using eXplainable Artificial Intelligence.” ScienceDirect. https://doi.org/10.1016/j.enbuild.2022.112521
3. Elgiriyewithana, Nidula. 2023. “Indian Weather Repository.” Kaggle. https://www.kaggle.com/datasets/nelgiriyewithana/indian-weather-repository-daily-snapshot
4. Greff, Klaus, Rupesh K. Srivastava, and Jan Koutnik. 2016. “LSTM: A Search Space Odyssey.” IEEE Xplore. https://ieeexplore.ieee.org/document/7508408
5. He, Renfei, Limao Zhang, and Alvin W. Chew. 2023. “Data-driven multi-step prediction and analysis of monthly rainfall using explainable deep learning.” ScienceDirect. https://doi.org/10.1016/j.eswa.2023.121160

6. Ioffe, Sergey, and Christian Szegedy. 2015. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.” Arxiv. https://arxiv.org/abs/1502.03167
7. Kim, Soyeon, Junho Choi, Yeji Choi, Subeen Lee, Artyom Stitsyuk, Minkyoung Park, Seongyeop Jeong, You-Hyun Baek, and Jaesik Choi. 2023. “Explainable AI-Based Interface System for Weather Forecasting Model.” Springer Nature Link. https://doi.org/10.1007/978-3-031-48057-7_7
8. Little, Roderick J. A., and Donald B. Rubin. 2002. *Statistical Analysis with Missing Data*. N.p.: Wiley.
9. Lundenberg, Scott, and Su-In Lee. 2017. “A Unified Approach to Interpreting Model Predictions.” Arxiv. https://arxiv.org/abs/1705.07874