Time Series Prediction of DC Precipitation by Using LSTM

Siru Wu
Data Science and Analytics
Georgetown University
Washington, D.C.
sw1430@georgetown.edu

Abstract

This study focuses on precipitation prediction in the Congressional region of Washington, D.C., a critical task for water management and drought monitoring. Using historical meteorological data from the ERA5 dataset, an LSTM-based time series model was developed to predict precipitation levels. Model performance was optimized through hyperparameter tuning and early stopping, with evaluation conducted using key metrics such as MSE, RMSE, and MAE. The results demonstrated the model's ability to capture precipitation trends while minimizing overfitting. Additionally, the environmental impact of model training was assessed using the CodeCarbon tool, highlighting its computational sustainability. These findings underscore the potential of LSTM models for enhancing precipitation prediction, offering valuable insights for climate-related policy and resource management.

1 Introduction

1.1 Background

Natural disasters cause extensive loss of life and damage to properties worldwide, with weather-related extremes intensifying due to climate change IPCC [2022]. Precipitation-driven events such as floods, droughts, and landslides present significant social and economic challenges, with their frequency and severity expected to increase Verdecchia et al. [2009] ;Banholzer et al. [2014]. Effective disaster risk forecasting depends on accurate hazard quantification, which is particularly complex for precipitation-related phenomena due to their inherent variability Blasone et al. [2024].

Precipitation prediction plays a critical role in water resource management, agricultural planning, and disaster mitigation, particularly in drought-prone regions like Washington, D.C. Accurate forecasts support policy-making and public safety by guiding resource allocation and emergency preparedness. However, the temporal and spatial complexity of precipitation data poses significant challenges for traditional prediction models.

1.2 Literature Review

Precipitation forecasting has historically been driven by statistical models such as Autoregressive Integrated Moving Average (ARIMA) and numerical weather prediction (NWP) frameworks NCE. While these methodologies have proven effective for short-term and localized weather forecasting, they encounter considerable challenges when applied to regions characterized by sparse and highly variable precipitation patterns, such as Washington, D.C. Recent advancements in deep learning architectures, particularly Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks Barzegar et al. [2020], have demonstrated significant improvements in predictive accuracy by capturing both spatial and temporal dependencies. However, despite these advancements, critical

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

research gaps persist, particularly concerning the integration of tailored models for specific climatic regions and the incorporation of sustainability considerations in deep learning-based meteorological forecasting.

1. **Lack of Precise Regional Modeling**: Most precipitation models are designed for large-scale global datasets, overlooking specific regional characteristics such as DC's unique topography and climatic variability Bader et al. [2008]. Tailored models for DC's specific conditions are underexplored.
2. **Limited Environmental Impact Studies**: Existing studies prioritize predictive accuracy while neglecting the carbon emissions associated with deep learning model training. The environmental cost of these models has become a pressing concern in AI sustainability.
3. **Data Sparsity and Imbalance**: Sparse data caused by irregular precipitation events poses challenges for deep learning models, especially during extreme weather conditions Stone [2004].
4. **Underexplored Green AI Practices**: Resource-efficient AI development remains limited in meteorological studies, despite its relevance to climate sustainability.

This study addresses these gaps by using an LSTM-based approach for DC-specific precipitation prediction, combined with a carbon footprint analysis using the CodeCarbon tool. By highlighting both predictive capabilities and environmental costs, this research aims to advance the development of environmentally sustainable AI models for meteorological applications.

2 Data

2.1 Data Overview

The dataset used in this study originates from the ERA5 reanalysis data, obtained through the Copernicus Climate Data Store (CDS) API CDS. The dataset encompasses meteorological measurements for the Washington, D.C. region, covering the time span from January to November 2024. The geographical coverage includes a narrowly defined area with coordinates ranging from 38.8977°N, -77.0365°E to 38.8975°N, -77.0363°E. The spatial resolution of approximately 11 km ($0.1° \times 0.1°$) ensures high-precision meteorological representation. Daily measurements are recorded at 12:00 PM local time, ensuring uniform sampling and relevance for midday weather conditions.

2.2 Feature Description

Total Precipitation (m): Represents the accumulated water depth from liquid and frozen precipitation (rain and snow) reaching the Earth's surface. It includes contributions from large-scale and convective precipitation events. The units are in meters, indicating the equivalent water depth if spread evenly across the grid cell.

2m Temperature (K): Air temperature measured at 2 meters above the Earth's surface. This temperature is interpolated from the lowest model level, accounting for near-surface atmospheric conditions. It is measured in Kelvin but can be converted to degrees Celsius by subtracting 273.15.

Surface Solar Radiation Downwards (J/m^2): The total amount of solar radiation reaching the Earth's surface, including both direct and diffuse components. This measurement reflects solar energy incident on the ground and is reported in joules per square meter (J/m^2). For energy flux representation, the accumulated values can be converted to watts per square meter (W/m^2) .

Soil Temperature Level 1 (K): Temperature of the soil's top layer (0 - 7 cm) measured in Kelvin. This metric is crucial for understanding heat transfer at the surface-soil interface and is essential for predicting precipitation and water retention processes Copernicus Climate Data Store [2024].

The precipitation prediction model relies on clearly defined input and target variables derived from the ERA5 dataset. The primary input feature, Total Precipitation, represents accumulated water depth from liquid and frozen precipitation, including rain and snow, reaching the Earth's surface. It accounts for both large-scale precipitation, generated by broad atmospheric systems, and convective precipitation, driven by localized atmospheric convection. Total Precipitation is measured in meters, indicating the equivalent water depth if uniformly distributed across the surface.

	Meteorological Variable	Level / Altitude	Unit
Input	Total Precipitation	Surface	m
	2m Temperature	2m	K
	Surface Solar Radiation Downwards	Surface	J/m²
	Soil Temperature Level 1	Surface/Soil Level 1	K
Target	Precipitation	Surface	m

Table 1: Meteorological variables, corresponding levels/altitudes, and units used in the study.

The target variable, referred to as Precipitation, represents the forecasted amount of precipitation expected at a future time step. Unlike Total Precipitation, which provides past accumulated values as historical context, the target precipitation corresponds to specific predicted future precipitation levels based on the model's learning process. This distinction ensures that past and future precipitation are separately considered, supporting accurate forecasting while adhering to best practices in predictive modeling.

3 Methodology

3.1 Model Architecture

This study employs a Long Short-Term Memory (LSTM)-based deep learning model to predict future precipitation levels in Washington, D.C. The meteorological variables serving as input features include total precipitation (tp), 2-meter temperature (t2m), surface solar radiation downwards (ssrd), and soil temperature at level 1 (stl1). Historical data for these variables is utilized to predict future precipitation levels.

LSTM (Long Short-Term Memory) networks are a type of Recurrent Neural Network (RNN) designed to capture long-term dependencies in sequential data. They consist of a cell state and three key gates—forget, input, and output—that manage information flow. At each time step t, the LSTM processes the current input x_t, the previous hidden state h_{t-1}, and the previous cell state c_{t-1}. The forget gate decides which information to discard, the input gate determines what new information to add, and the cell state is updated accordingly. Finally, the output gate generates the current hidden state h_t, which is used for predictions or passed to the next time step. This design allows LSTMs to effectively learn complex patterns over time [Thomas, 2024].

In this study, the LSTM model architecture consists of a two-layer stacked LSTM structure, with each layer containing 256 hidden units. To mitigate the risk of overfitting, a 30% dropout rate is applied after each LSTM layer. The output layer is a fully connected layer that maps the outputs of the LSTM layers to the target variable—future precipitation levels. Both input data and target variables are standardized to enhance training stability and accelerate convergence. This transformation ensures that each feature has a mean of 0 and a standard deviation of 1, expediting the training process and improving model performance.

The dataset is split into training (80%), validation (15%), and testing (5%) sets. The validation set is used to evaluate the model's performance during training, helping to prevent overfitting through real-time monitoring. During training, the AdamW optimizer is employed, and Mean Squared Error (MSE) is selected as the loss function to optimize model parameters via backpropagation. To further address overfitting, an early stopping mechanism halts the training process if validation loss does not improve significantly for 10 consecutive epochs.

The time step is set to 5, meaning the model uses data from the previous five time points to predict the next precipitation value. This configuration enables the model to capture patterns over an extended time range. The training process spans 200 epochs, during which the model iteratively adjusts its weights based on the computed error to optimize prediction accuracy.

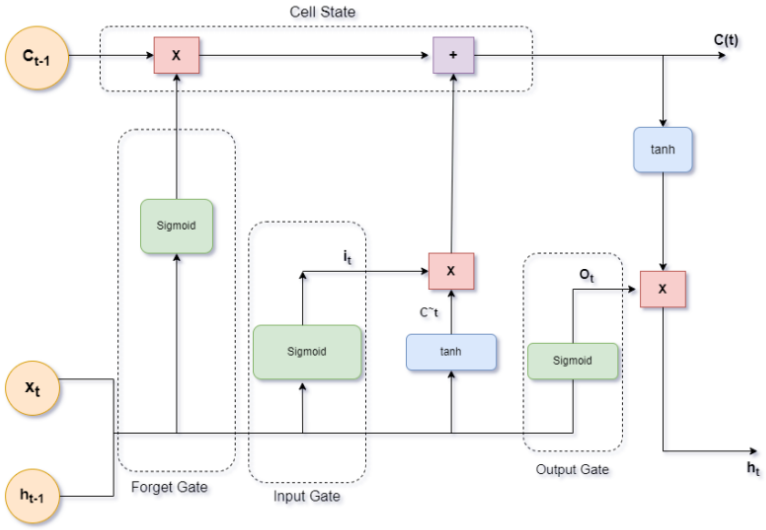

Figure 1: LSTM Model Architecture for Precipitation PredictionThomas [2024]

3.2 Evaluation Metrics

After training, the model is evaluated on the test set, with predictions being inverse-transformed to the original data scale through de-standardization. To comprehensively evaluate the model's predictive performance, several metrics are used, including Mean Absolute Error (**MAE**), Root Mean Squared Error (**RMSE**), Mean Absolute Percentage Error (**MAPE**), and the coefficient of determination ($\mathbf{R^2}$).

The Mean Absolute Error (**MAE**) measures the average magnitude of prediction errors, calculated as:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

The Root Mean Squared Error (**RMSE**) evaluates the standard deviation of prediction errors, calculated as:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

The Mean Absolute Percentage Error (**MAPE**) indicates prediction accuracy as a percentage, computed as:

$$\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

The coefficient of determination (R^2) assesses the proportion of variance in the target variable explained by the model, given by:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

where $\bar{y}$ is the mean of the actual values.

3.3 Carbon Emissions Tracker

In this study, the **CodeCarbon** tool was employed to monitor and estimate the environmental impact of the LSTM model training process. CodeCarbon is an open-source library that provides real-time tracking of carbon emissions by calculating energy consumption and associating it with the carbon intensity of the electricity grid at the computational resource's location.

Carbon Intensity of the consumed electricity is calculated as a weighted average of the emissions from the different energy sources that are used to generate electricity, including fossil fuels and renewables. In this toolkit, the fossil fuels coal, petroleum, and natural gas are associated with specific carbon intensities: a known amount of carbon dioxide is emitted for each kilowatt-hour of electricity generated. Renewable or low-carbon fuels include solar power, hydroelectricity, biomass, geothermal, and more. The nearby energy grid contains a mixture of fossil fuels and low-carbon energy sources, called the Energy Mix. Based on the mix of energy sources in the local grid, the Carbon Intensity of the electricity consumed can be computed[CodeCarbon, 2024].

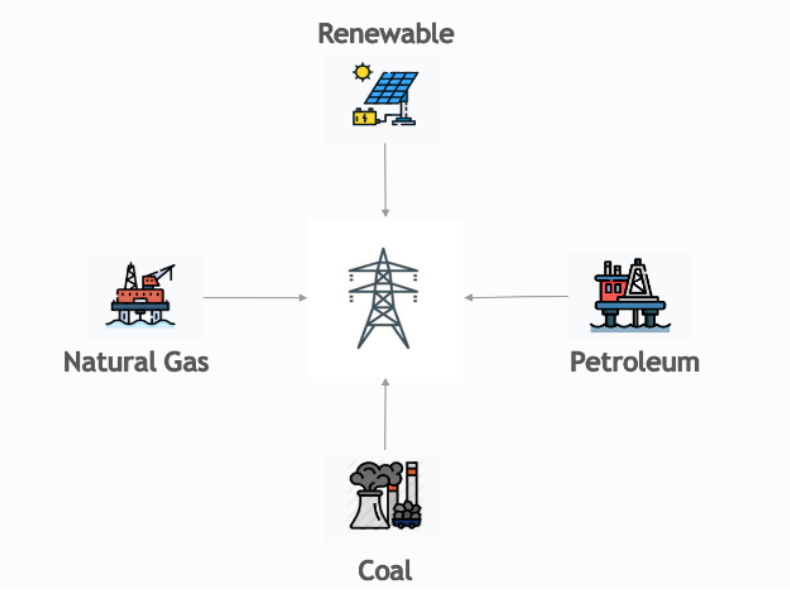

Figure 2: The energy grid comprises various energy sources, including renewable energy, natural gas, petroleum, and coal.

CodeCarbon calculates carbon emissions using the formula:

$$E = P \times T \times \text{EF}$$

where E represents the total carbon emissions (in kilograms of CO_2), P is the average power consumption during training (in kilowatts), T is the total runtime of the model (in hours), and EF is the emission factor of the electricity grid, which reflects the amount of CO_2 emitted per kilowatt-hour of electricity consumed. The emission factor is location-dependent, and CodeCarbon automatically retrieves this value based on the IP address or cloud region of the computational resources[CodeCarbon, 2024].

Energy consumption (EC) was computed as:

$$EC = P \times T$$

where P is measured in watts and converted to kilowatts to align with the standard energy measurement conventions. This accounts for the total energy used during the training process.

CodeCarbon also incorporates the Power Usage Effectiveness (PUE) metric, which measures the efficiency of data centers by accounting for energy overheads such as cooling and additional infrastructure. This adjustment ensures that the reported energy consumption reflects the total environmental cost of the training process.

Throughout the training process, CodeCarbon logged energy usage and carbon emissions data in real time, providing a detailed record of the environmental footprint of the computation. This integration emphasizes the importance of sustainability in machine learning workflows, ensuring that model development aligns with principles of Green AI.

4 Results and Analysis

4.1 Evaluation Metrics Result

The performance of the model was assessed using four key evaluation metrics: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE), and the Coefficient of Determination (R^2).

Metric	Value
MAE	0.301114
RMSE	0.194423
MAPE	128.111729%
R^2	0.617230

Table 2: Model Performance Metrics

The **MAE** of 0.301114 reflects that, on average, the predicted precipitation values deviate from the actual values by approximately 0.30. This result indicates a moderate level of accuracy, suggesting that the model performs well in capturing general precipitation patterns. However, some deviations persist, particularly in regions with pronounced variability in precipitation.

The **RMSE** of 0.194423 highlights the model's capability to minimize larger prediction errors. By emphasizing squared differences, this metric indicates that the model fits the data effectively in most cases, although certain regions with rapid or irregular precipitation fluctuations may still present challenges.

The **MAPE** of 128.11% indicates significant relative errors in scenarios with low precipitation or extreme variability. This elevated MAPE reflects the model's difficulty in handling outliers or sparse data, where predictions deviate more significantly from observed values.

The $\mathbf{R^2}$ score of 0.617230 shows that the model explains approximately 62% of the variance in precipitation levels. While this demonstrates a good ability to capture overall precipitation trends, a considerable portion of variability remains unaccounted for, highlighting the complexity of the precipitation patterns in the dataset.

The performance metrics illustrate the model's strengths in capturing general trends and minimizing larger errors. However, the high MAPE and the unexplained variability reflected in the $\mathbf{R^2}$ score emphasize challenges in regions with extreme precipitation variability or sparse data. These findings suggest opportunities for further improvement, such as incorporating additional meteorological features or exploring advanced model architectures, to enhance predictive performance and better address complex precipitation patterns.

4.2 Analysis of Model Training

The training and validation loss curves (Figure 3) provide insights into the model's learning process over the course of training. The x-axis represents the number of epochs, where each epoch corresponds to a complete pass through the training data. The y-axis shows the loss values in meters, which quantify the prediction error of the model. The red curve represents the training loss, while the blue curve represents the validation loss.

As the number of epochs increases, the training loss decreases consistently, indicating that the model effectively learns patterns from the training data. Similarly, the validation loss shows a significant reduction during the initial epochs and stabilizes in the later stages, reflecting improved generalization on unseen data. However, the persistent gap between the training and validation losses suggests a slight overfitting tendency, where the model performs better on the training set compared to the validation set. The absence of a sharp increase in validation loss indicates that severe overfitting was

successfully avoided, and the model demonstrates a reasonable balance between fitting the training data and generalizing to new samples.

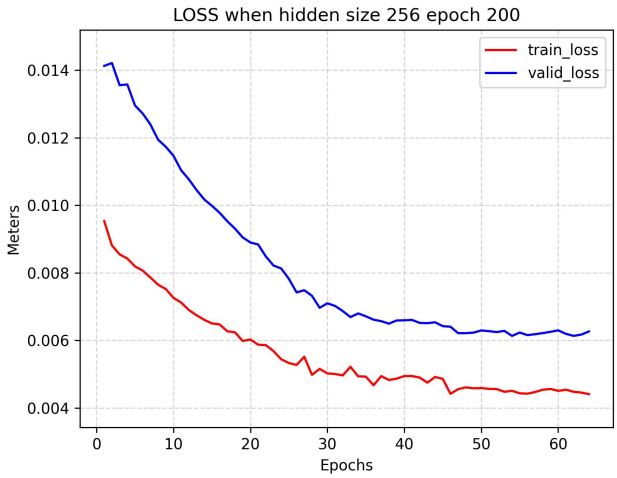

Figure 3: Loss curve showing training and validation loss over epochs

4.3 Analysis of Prediction Performance

The prediction results (Figure 4) compare the actual precipitation values (red curve) with the predicted values (blue curve) across a time series. The x-axis represents time in hours, where each increment corresponds to a 12-hour period, while the y-axis indicates precipitation levels in meters. The graph illustrates that the model performs well in regions of low precipitation, where the predicted values align closely with the actual values. However, discrepancies are evident in high-variability regions, particularly during periods of extreme precipitation. In these cases, the predicted values often underestimate the magnitude of high precipitation peaks, as observed around the 1000–1200 and 2000–2500 hour intervals. This indicates the model's difficulty in accurately capturing short-term fluctuations and extreme precipitation events. Nevertheless, the overall trend of the predicted values follows the pattern of the actual data, showcasing the model's ability to capture the general temporal dynamics of precipitation.

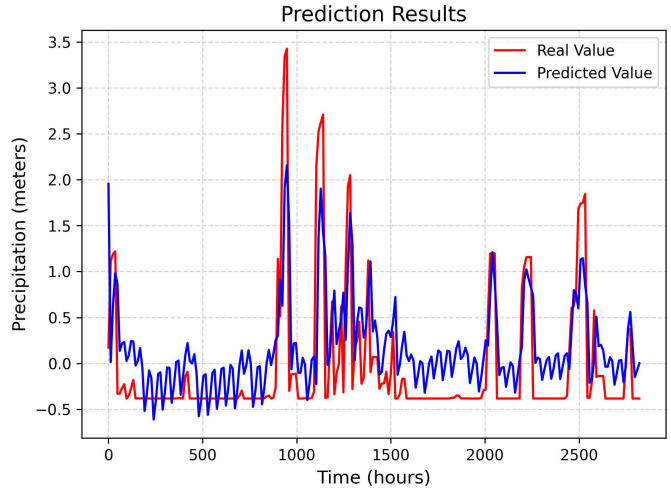

Figure 4: Prediction results comparing real and predicted precipitation values over time

In conclusion, the loss curve analysis suggests that the model effectively minimizes prediction errors during training and generalizes reasonably well to validation data, although some overfitting tendencies persist. Meanwhile, the prediction results demonstrate the model's strength in capturing general precipitation trends but highlight its limitations in accurately predicting extreme events.

4.4 Environmental Efficiency of Model Training

During this model training, the total training duration was approximately 28.27 seconds. The total carbon emissions amounted to 0.000131475 kg CO_2 (equivalent to 0.131475 grams), indicating a minimal environmental impact. The energy consumption for the entire training process was 0.00035617 kWh, reflecting efficient electricity usage. The emissions rate during training was recorded at 4.65E-06 kg CO_2 per second, and the CPU energy expenditure was measured at 0.000333761 kWh, highlighting the primary source of energy utilization. The power consumption during training was 42.5 watts, demonstrating moderate power demand throughout the process.

Metric	Value
Duration (seconds)	28.27
Carbon Emissions (kg CO_2)	0.000131475
Emissions Rate (kg CO_2/second)	4.65E-06
Total Energy Consumed (kWh)	0.00035617
CPU Energy Consumed (kWh)	0.000333761
Power Consumption (Watts)	42.5

Table 3: Environmental Impact Metrics of Model Training

By monitoring carbon emissions and energy usage in real-time, the environmental impact of computational processes can be quantitatively assessed, enabling precise evaluation of sustainability in machine learning workflows. These findings underscore that while the training process demands computational resources, it remains efficient and contributes to environmentally conscious development practices.

5 Limitations

While the model demonstrated its capability to predict precipitation trends, several limitations emerged during the evaluation. The most notable limitation lies in its difficulty capturing seasonal and periodic variations inherent in precipitation data. Seasonal patterns, characterized by periodic fluctuations, often lead to deviations in the model's predictions, particularly during extreme weather conditions or transitions between seasons, such as summer and winter. Although the LSTM model effectively captures general trends, its architecture lacks explicit mechanisms to model such periodic variations, reducing its predictive accuracy during these periods.

Another limitation stems from the meteorological variables used in the dataset. While the model incorporated key features such as total precipitation, 2-meter temperature, surface solar radiation, and soil temperature, it excluded other critical factors like wind speed, air pressure, and humidity, which are known to influence precipitation patterns. This omission may result in the model overlooking important interactions and dependencies, limiting its comprehensiveness. Furthermore, the reliance on the ERA5 dataset, although high-resolution, restricts the model's ability to account for localized weather phenomena that can substantially impact precipitation.

These limitations underscore the need to refine the model architecture and expand the dataset to improve its capacity to address seasonal and complex precipitation patterns comprehensively.

6 Future Works

To address the identified limitations, future research should focus on enhancing the model's capacity to handle seasonal variations and improving the dataset's comprehensiveness. Specialized techniques, such as seasonal decomposition or Fourier Transform, could be employed to extract seasonal components from the data Lau and Weng [1995]. Integrating these components into the model as explicit

features would enable it to better capture periodic fluctuations and improve its performance during seasonal transitions.

Exploring alternative models and hybrid approaches may also yield significant improvements. For example, Gated Recurrent Units (GRUs) could offer a more efficient solution for handling dependencies Pan et al. [2020], while Transformer models, with their self-attention mechanisms, can effectively capture both long-term dependencies and periodic variations Kim and Kim [2022]. Hybrid architectures combining LSTMs with Convolutional Neural Networks (CNNs) or attention mechanisms could further enhance the model's ability to focus on critical temporal segments Barzegar et al. [2020].

Expanding the dataset to include additional meteorological variables such as wind speed, humidity, and air pressure would provide a more comprehensive representation of the factors influencing precipitation. Incorporating diverse data sources, such as satellite observations or ground-based measurements, could also enrich the dataset and improve the model's robustness against varying conditions.

Finally, refining the data preprocessing pipeline could significantly boost the model's predictive accuracy. Advanced feature engineering techniques, such as temporal markers for seasonal or holiday effects, and enhanced normalization or standardization methods could optimize the quality of input data. These steps, combined with a focus on hyperparameter optimization and scalable training methods, would ensure that the model is better equipped to handle the complexities of precipitation prediction, paving the way for more reliable and accurate results.

7 Conclusion

This study explored the application of Long Short-Term Memory (LSTM) networks for precipitation predictioning in Washington, D.C., providing valuable insights into both the potential and challenges of using deep learning in meteorological prediction. The results demonstrated that while the LSTM model effectively captured general precipitation patterns, its performance declined in regions with high variability or extreme weather events. This indicates that while advanced neural network models can support weather prediction, addressing complex meteorological dynamics requires further enhancements.

Key findings revealed that incorporating meteorological features such as temperature, solar radiation, and soil temperature enabled the model to make meaningful predictions, although the exclusion of additional variables like wind speed or air pressure limited its predictive accuracy. Moreover, the study highlighted the critical role of environmental sustainability in machine learning research. By tracking carbon emissions and energy consumption during model training, this research underscored the feasibility of balancing computational demands with sustainability goals.

The findings from this research have practical implications for environmental monitoring, water resource management, and disaster preparedness. More accurate precipitation predictions can inform policy decisions, improve agricultural planning, and enhance public safety measures. As climate change continues to intensify, advancing predictive technologies can support proactive climate resilience strategies.

Looking ahead, integrating more diverse datasets, optimizing model architectures, and incorporating sustainability measures will further improve prediction performance. This study contributes to the broader field of climate informatics, bridging data science and environmental science to promote technological innovation for real-world impact.

References

Cds. https://cds.climate.copernicus.eu/.

National centers for environmental information (ncei). https://www.ncei.noaa.gov/products/weather-climate-models/numerical-weather-prediction.

David Bader, Curt Covey, William Gutowski, Isaac Held, Kenneth Kunkel, Ronald Miller, Robin Tokmakian, and Minghua Zhang. Climate models: an assessment of strengths and limitations. 2008.

Sandra Banholzer, James Kossin, and Simon Donner. *The Impact of Climate Change on Natural Disasters*, pages 21–49. 01 2014. ISBN 978-94-017-8597-6. doi: 10.1007/978-94-017-8598-3_2.

Rahim Barzegar, Mohammad Taghi Aalami, and Jan Adamowski. Short-term water quality variable prediction using a hybrid cnn–lstm deep learning model. *Stochastic Environmental Research and Risk Assessment*, 34(2):415–433, 2020.

Valentina Blasone, Erika Coppola, Guido Sanguinetti, Viplove Arora, Serafina Di Gioia, and Luca Bortolussi. A deep learning framework to efficiently estimate precipitation at the convection permitting scale. In *Workshop on Tackling Climate Change with Machine Learning, ICLR*. ICLR, 2024.

CodeCarbon. Codecarbon methodology. Online, 2024. URL https://mlco2.github.io/codecarbon/methodology.html. Accessed: December 3, 2024.

Copernicus Climate Data Store. ERA5-Land: Reanalysis Dataset Overview. https://cds.climate.copernicus.eu/datasets/reanalysis-era5-land?tab=overview, 2024. Accessed: 2024-12-13.

IPCC. *Climate Change 2022: Impacts, Adaptation and Vulnerability. Summary for Policymakers*. Cambridge University Press, Cambridge, UK and New York, USA, 2022. ISBN 9781009325844.

Dong-Keon Kim and Kwangsu Kim. A convolutional transformer model for multivariate time series prediction. *IEEE Access*, 10:101319–101329, 2022. doi: 10.1109/ACCESS.2022.3203416.

K.-M. Lau and Hengyi Weng. Climate signal detection using wavelet transform: How to make a time series sing. *Bulletin of the American Meteorological Society*, 76(12):2391 – 2402, 1995. doi: 10.1175/1520-0477(1995)076<2391:CSDUWT>2.0.CO;2. URL https://journals.ametsoc.org/view/journals/bams/76/12/1520-0477_1995_076_2391_csduwt_2_0_co_2.xml.

Mingyang Pan, Hainan Zhou, Jiayi Cao, Yisai Liu, Jiangling Hao, Shaoxi Li, and Chi-Hua Chen. Water level prediction model based on gru and cnn. *IEEE Access*, 8:60090–60100, 2020. doi: 10.1109/ACCESS.2020.2982433.

Peter H Stone. Climate prediction: The limits of ocean models. 2004.

Babu Thomas. Understanding lstm: An in-depth look at its architecture, pros, and cons. Online, 2024. URL https://www.linkedin.com/pulse/understanding-lstm-in-depth-look-its-architecture-pros-babu-thomas/. Accessed: December 3, 2024.

Marco Verdecchia, Erika Coppola, Barbara Tomassetti, and Guido Visconti. Cetemps hydrological model (chym), a distributed grid-based model assimilating different rainfall data sources. 2009. URL https://api.semanticscholar.org/CorpusID:126518829.