Summer Research Fellowship Programme

Report On
LSTM Variants for Short-Term Temperature and Precipitation Forecasting: A Comparative Study

Al Faiz Ali, ENGS1066,
Indian Institute of Engineering Science and Technology, Shibpur

UNDER THE SUPERVISION OF
Prof. V. Jothiprakash
Indian Institute of Technology Bombay
July 2025

Acknowledgment

First and foremost, I would like to express my deepest gratitude to the Indian Academy of Sciences (IAS), Indian National Science Academy (INSA), and The National Academy of Sciences, India (NASI) for selecting me as a Summer Research Fellow (2025). This prestigious opportunity through the Summer Research Fellowship Programme (SRFP) has been instrumental in advancing my research capabilities in environmental forecasting and deep learning applications.

I am deeply grateful to my mentor, Prof. V. Jothiprakash, Department of Civil Engineering, Indian Institute of Technology Bombay, for his exceptional guidance, consistent encouragement, and unwavering support throughout the course of this project. His insightful feedback and mentorship played a crucial role in shaping the direction, quality, and depth of this work. I feel privileged to have learned under his guidance.

I would also like to express my heartfelt thanks to Ms. Injila Hamid, Ph.D. scholar at IIT Bombay, for her invaluable support in providing the IMDAA reanalysis dataset for the Vaitarna River Basin and for offering continuous technical assistance during data preparation, modeling, and analysis. Her clarity in explaining complex hydrological concepts and her readiness to help at every step significantly enhanced my understanding and learning experience.

My sincere thanks to Mr. Digvijay Singh, M.Tech. student at IIT Bombay, for sharing relevant research papers and resources that greatly contributed to the literature review and methodology design sections of this project.

This internship has given me a unique platform to work on real-world applications of LSTM models for temperature and precipitation forecasting. It has strengthened my academic foundation and introduced me to practical aspects of time series modeling and hydrological data analysis. I am confident that the skills, insights, and experiences gained during this internship will serve as a strong base for my future academic and professional endeavors.

Lastly, I extend my gratitude to my family and peers for their constant support, encouragement, and belief in my potential, which kept me motivated throughout this journey. This experience will always remain a valuable milestone in my academic career.

Abstract

This project focuses on forecasting temperature and precipitation using Long Short-Term Memory (LSTM) models. The goal was to predict future values one hour ahead (single-step) and several hours ahead (multi-step) using historical weather data. The data was collected from multiple years, cleaned, and prepared by handling missing values, scaling features, and reshaping it for LSTM input.

To understand the patterns in the data, we performed time series analysis including trend and seasonality decomposition, autocorrelation checks, and False Nearest Neighbor (FNN) analysis. The FNN analysis helped determine how many past time steps (lags) should be used as input for the models. We trained and compared three types of LSTM models: Vanilla, Stacked, and Bidirectional on both temperature and precipitation datasets. Additional features like lag values, rolling averages, and time-based columns (such as hour, day, and year) were incorporated to enhance the model's learning capability.

The models were evaluated using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared value. For single-step temperature forecasting, the Bidirectional LSTM produced the best results, while for multi-step temperature prediction, the Vanilla LSTM demonstrated superior performance with lower validation error and strong generalization. In the case of precipitation forecasting, the Stacked LSTM outperformed both Vanilla and Bidirectional variants in both single-step and multi-step tasks, offering more stable and consistent performance despite the irregular nature of rainfall data. All trained models were saved along with their respective scalers to facilitate future predictions using the latest available data.

This project demonstrates that LSTM models, when supported by appropriate data preprocessing, feature engineering, and sequence design techniques such as False Nearest Neighbors (FNN), can be effectively employed for short-term environmental forecasting. The methodology and findings lay a solid foundation for building practical, operational weather prediction systems that can benefit a range of applications requiring accurate and timely forecasts of temperature and precipitation.

Contents

1 Introduction

Accurate short-term forecasting of environmental variables like temperature and precipitation is essential for weather monitoring, disaster management, agriculture, and urban planning. With the increasing availability of historical weather data, machine learning and deep learning models have become valuable tools for time series forecasting.

This project focuses on using Long Short-Term Memory (LSTM) neural networks to forecast temperature and precipitation values for both single-step (one hour ahead) and multi-step (several hours ahead) time horizons. LSTM is well-suited for this task due to its ability to learn long-term dependencies in sequential data, making it effective for modeling the temporal patterns present in weather time series.

The work involves collecting multi-year univariate datasets from the Indian Monsoon Data Assimilation and Analysis reanalysis (IMDAA), which include hourly temperature and precipitation records. These datasets are cleaned, merged, and preprocessed by handling missing values, converting timestamps, and scaling features. To better understand the structure of the time series, exploratory analysis is performed including trend and seasonality decomposition, autocorrelation studies, and the Augmented Dickey-Fuller (ADF) test for stationarity.

Additionally, the False Nearest Neighbour (FNN) method is applied to help in understanding how many past observations are needed to adequately represent the system's dynamics, which is critical for building effective forecasting models.

Three LSTM variants namely Vanilla, Stacked, and Bidirectional, are designed, trained, and evaluated on both temperature and precipitation datasets. Each model's performance is assessed using standard error metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and the Coefficient of Determination (R^2), and the trained models are saved for future use.

This project aims to identify the most suitable LSTM architecture for environmental forecasting tasks and create a reliable framework that can be extended to other similar time series problems.

2 Literature Review

The Long Short-Term Memory (LSTM) model was originally proposed by **Hochreiter and Schmidhuber (1997)** to address the vanishing gradient problem in standard Recurrent Neural Networks (RNNs), enabling effective learning of long-range temporal dependencies in time series data.

In the domain of hydrology and flood forecasting, numerous studies have explored the capabilities of LSTM and its variants. For instance, **Kratzert et al. (2018)** demonstrated that LSTM-based rainfall–runoff models outperformed traditional conceptual hydrological models across several catchments.

Liu et al. (2023) applied LSTM to forecast short-term water levels in the Xiangjiang River. Their study showed that LSTM produced better RMSE and NSE metrics compared to GRU and vanilla RNN models, especially for 6-hour and 12-hour flood forecasts, confirming its robustness in capturing nonlinear water-level dynamics.

In another study, **Jhong et al. (2024)** developed a hybrid model combining Genetic Algorithms (GA) with LSTM for flash flood forecasting in the Wu River, Taiwan. This model fine-tuned LSTM parameters such as hidden neurons and learning rate using GA, leading to improved accuracy (NSE values of 0.917 to 0.931), especially for short lead-time predictions.

Garg et al. (2023) focused on the Godavari River basin and created a multivariate, multi-step LSTM model for runoff prediction using multiple hydrometeorological inputs. Their approach showed accurate multi-horizon predictions and proved useful for basin-level flood management.

Fang et al. (2021) employed LSTM for flood susceptibility mapping in the Poyang Lake Basin. By incorporating environmental and hydrological data, their LSTM model achieved higher spatial prediction accuracy compared to traditional GIS and machine learning methods.

Moon et al. (2023) proposed a hybrid model that combined physical hydrological models with LSTM-based deep learning for urban flood prediction. This approach leveraged both domain knowledge and data-driven learning to enhance urban flood response strategies.

These studies underscore the growing reliance on LSTM and its variants in flood and hydrological forecasting, due to their flexibility, accuracy, and capacity to model nonlinear, time-dependent processes across different temporal and spatial scales.

Table 1: Summary of Literature Review on LSTM Applications in Environmental Forecasting

Year	Author(s)	Method	Application	Key Contribution
1997	Hochreiter & Schmidhuber	LSTM	General time-series modeling	Solved vanishing gradient problem in RNNs; foundational architecture for sequence modeling with memory components
2021	Fang et al.	LSTM	Flood susceptibility mapping	Proposed LSS-LSTM; improved spatial modeling of flood-prone areas using feature engineering and spatial sequences
2023	Moon, Yoon & Moon	Hybrid (Rainfall-runoff + LSTM)	Urban flood forecasting	Combined physical runoff models with LSTM to forecast flooding time from rainfall scenarios in urban basins
2023	Garg et al.	Multivariate Multi-step LSTM	Flood runoff prediction (Godavari Basin)	Built LSTM model using 8 hydrometeorological variables; achieved strong accuracy for 1-day and 7-day runoff prediction
2023	Liu et al.	LSTM	River flood forecasting (Xiangjiang River)	Evaluated LSTM against RNN & GRU; best accuracy for short-term flood levels; stressed role of input features
2024	Jhong et al.	GA + LSTM	Flash flood forecasting	Used Genetic Algorithm to optimize LSTM parameters

3 Dataset Description

This study uses meteorological data provided by the Indian Monsoon Data Assimilation and Analysis reanalysis (IMDAA) for the Vaitarna River Basin region in Maharashtra, India. The dataset serves as the foundational input for building deep learning models aimed at forecasting two key environmental variables: temperature and precipitation.

3.1 Variables and Data Structure

The dataset includes two core variables: temperature, measured in degrees Celsius (°C), and precipitation, measured in millimeters (mm). Both variables are recorded at an hourly frequency, providing fine-grained temporal resolution that is ideal for short-term forecasting using LSTM models. This high-frequency nature enables the models to capture rapid changes in weather, particularly useful for predicting extreme rainfall or temperature fluctuations.
Each file originally included only the temperature and precipitation values. The date-time column was manually added later during preprocessing using appropriate timestamp formatting to enable time series operations and sequencing.
Each yearly folder contains subfiles corresponding to different spatial grid points. These grid points are identified by their latitude and longitude coordinates. There are a total of 8 unique latitude values and 10 unique longitude values, resulting in a grid of 80 spatial locations (8×10). Each CSV file within a folder represents data from one specific grid point for that particular year.

3.2 Temporal and Spatial Coverage

The dataset spans a period from 1996 to 2019, covering over two decades of hourly meteorological observations. The data is organized in folders, with each folder representing one year of data. Within each folder, there are 80 CSV files, corresponding to different grid locations based on latitude and longitude.
For this study, a single grid location (Latitude: 19.32, Longitude: 72.72000000000003) was selected for modeling. During preprocessing, the files corresponding to this specific grid point across all years were looped through and concatenated chronologically to create a continuous time series.

3.3 Data Quality and Completeness

Upon initial inspection, the dataset was found to be of high quality, with no missing values. As a precautionary step, data completeness checks were performed, and

missing timestamps or readings (if any) were filled using linear interpolation to preserve temporal continuity. This step is crucial for maintaining the integrity of sequential learning models like LSTMs.
Additionally, care was taken to sort each dataset chronologically and ensure all timestamps were uniformly spaced at hourly intervals. This consistent time structure is essential for training time series models effectively.

4 Data Preprocessing

Preprocessing is a crucial stage in any time series forecasting task, as it ensures the raw data is clean, structured, and transformed into a format suitable for model training. For this project, the preprocessing steps were applied separately but using a similar methodology to both temperature and precipitation datasets. The data from the Indian Monsoon Data Assimilation and Analysis reanalysis (IMDAA) was hourly and spanned multiple years and locations within the Vaitarna basin.

4.1 Data Loading and Cleaning

The initial preprocessing involved reading and merging data from multiple files organized by year and spatial location (latitude-longitude grid). Each file originally contained only the temperature or precipitation column. A timestamp column was added manually to all files to serve as a common time index.
The steps were as follows:
Parsing Timestamps: The manually added time column was converted into Python datetime format and set as the index for the DataFrame. This step ensures correct chronological ordering and enables time-based slicing and resampling.
Concatenation: For the selected spatial location (identified by Latitude: 19.32, Longitude: 72.72000000000003), data files across all available years (1996–2019) were concatenated in chronological order. This produced a single continuous time series per variable (temperature or precipitation) for that grid point.
Sorting: The data was sorted by timestamp to ensure chronological consistency across all records.
Location-Specific Preprocessing: While the dataset includes 80 spatial grid locations, modeling and analysis in this study were conducted on only one selected grid. All preprocessing, feature engineering, and model training steps were applied specifically to this location, enabling the model to learn localized temporal patterns. This approach ensures that the weather dynamics at the chosen location are modeled accurately, which is particularly important for hydrological applications where temperature and precipitation patterns vary significantly across regions.

4.2 Handling Missing Values

Both the temperature and precipitation datasets were found to be complete, with no missing values detected during the initial data quality checks. However, as a precautionary step to ensure robustness and consistency in the preprocessing pipeline, missing value handling techniques were still incorporated.
Specifically, linear interpolation was applied using pandas.interpolate(method='linear'). Although no gaps were present, this step safeguarded against any potential edge cases or hidden inconsistencies, while preserving the continuity and underlying temporal patterns of the data.
In the context of time series forecasting, interpolation helps prevent disruptions in the sequential input required by LSTM models, and does so without introducing artificial trends or abrupt shifts. This makes it a safe and effective fallback strategy even when the dataset appears complete.

4.3 Exploratory Time Series Analysis

Exploratory analysis was carried out to gain a deeper understanding of the underlying structure and temporal patterns present in both the temperature and precipitation datasets. Time series decomposition was applied to identify long-term trends and recurring seasonal components such as daily and annual cycles. This decomposition provided valuable insight into the distinct temporal behaviors of the two variables, with temperature exhibiting clear seasonality and trends, while precipitation remained more irregular.
Autocorrelation analysis was conducted using Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots to examine the relationships between current observations and their past values. The temperature series displayed strong autocorrelation patterns, especially at daily and seasonal lags, reflecting its smooth and periodic nature. In contrast, the precipitation series exhibited much weaker autocorrelation, consistent with its sporadic and event-driven characteristics.
To assess the statistical stability of the time series, Augmented Dickey-Fuller (ADF) tests were performed for both variables. Although LSTM models do not explicitly require stationarity, evaluating stationarity helps in understanding the variability of the data over time and supports the design of appropriate features. Overall, this exploratory analysis informed key preprocessing and modeling decisions.

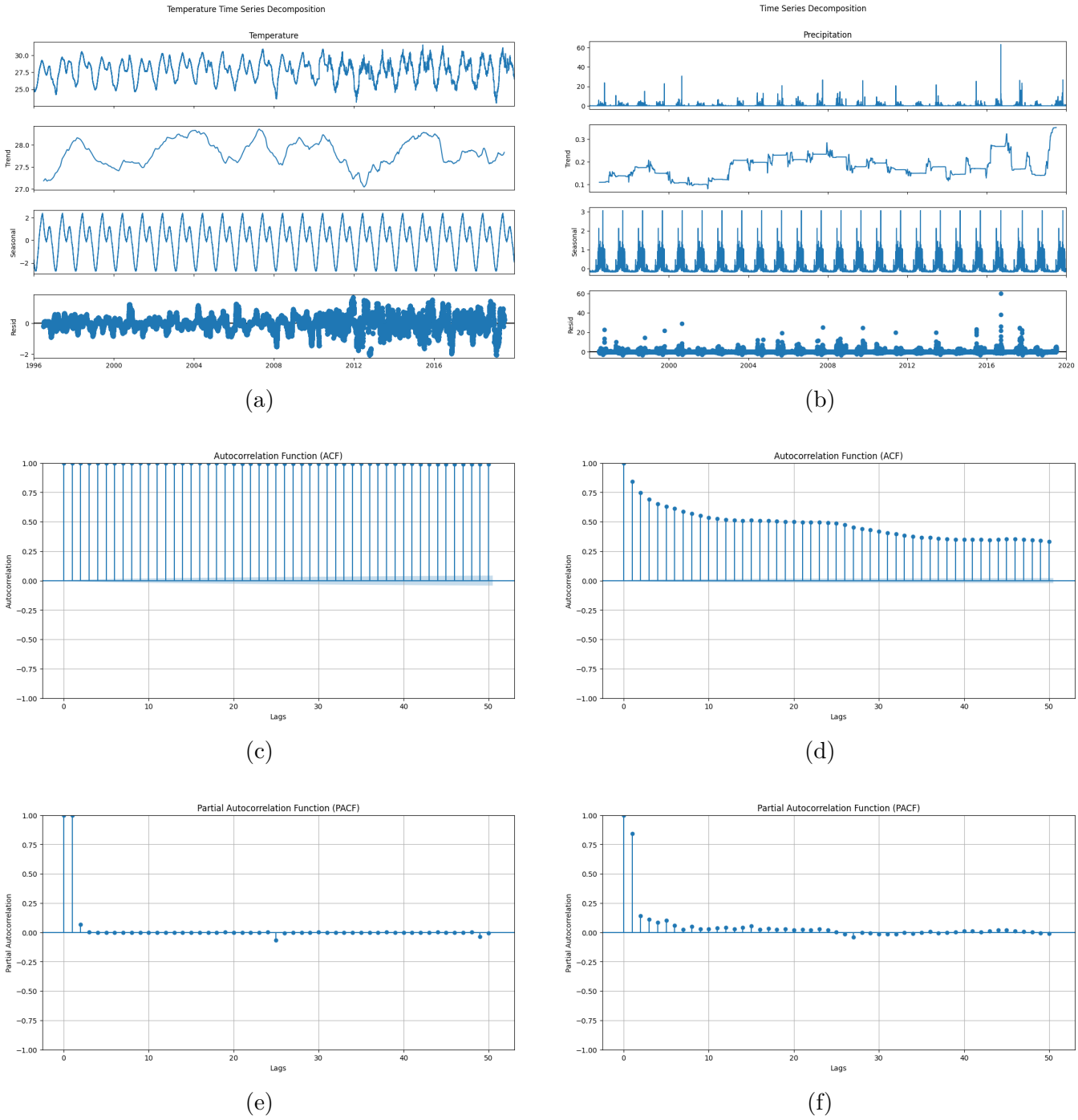


Figure 1: Time series analysis results showing: (a) Temperature time-series decomposition, (b) Temperature autocorrelation analysis (ACF), (c) Precipitation time-series decomposition, (d) Precipitation autocorrelation analysis (ACF), (e) Temperature partial autocorrelation analysis (PACF), and (f) Precipitation partial autocorrelation analysis (PACF). The decomposition plots reveal trend and seasonality components, while ACF/PACF analyses demonstrate temporal dependencies in both temperature and precipitation datasets.

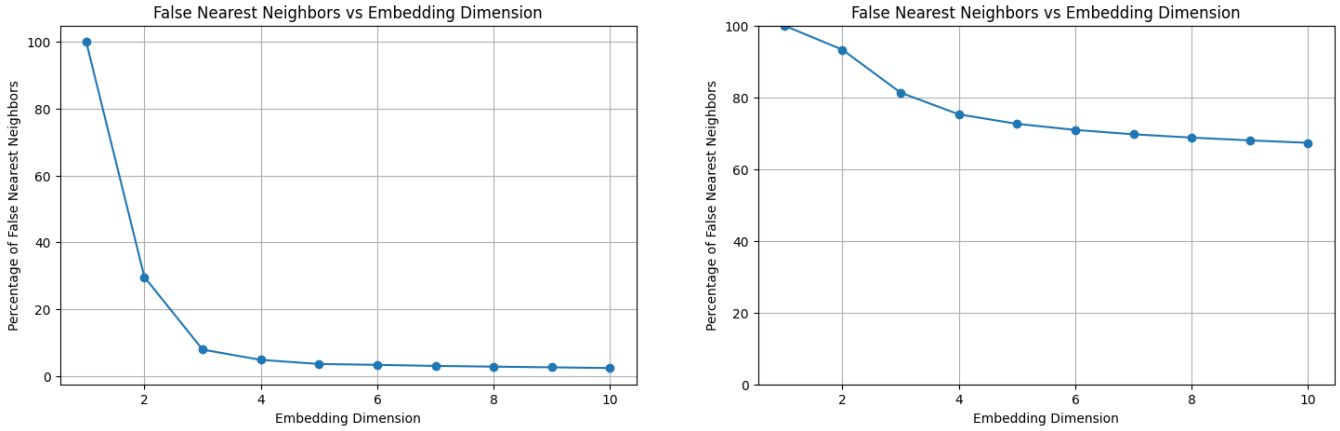
4.4 False Nearest Neighbors (FNN) Analysis

False Nearest Neighbors (FNN) analysis was conducted to estimate the optimal embedding dimension, which refers to how many past values are required to adequately reconstruct the system's state for time series forecasting.

For the temperature dataset, the FNN curve showed a clear drop in false neighbor percentages with increasing dimensions, suggesting an optimal embedding dimension of $m = 3$ with a time delay $\tau = 24$. This aligns well with the choice of a 24-hour input window used during LSTM model training. The structured periodic behavior and strong autocorrelation in temperature time series made FNN a reliable tool in guiding sequence length selection.

In contrast, for the precipitation data, the FNN analysis did not yield a distinct minimum in the curve. The false nearest neighbor ratio remained high across all tested dimensions, indicating weak deterministic structure and high randomness in the rainfall patterns. This made it difficult to infer an optimal embedding dimension from the FNN method. As a result, the input sequence length for precipitation modeling was selected empirically, based on iterative testing and validation performance.

The corresponding FNN plots for both temperature and precipitation data are shown in Figure **??**, illustrating the contrasting behaviors and justifying the different approaches taken in defining input window lengths.



(a) (b)

Figure 2: False Nearest Neighbors analysis for (a) temperature and (b) precipitation datasets. The temperature analysis shows clear deterministic structure with an optimal embedding dimension of 3 and time delay (τ) of 24 hours. In contrast, the precipitation analysis reveals stochastic behavior with no clear minimum dimension and consistently high FNN% values across all tested dimensions, reflecting the unpredictable nature of rainfall patterns.

4.5 Feature Engineering

Feature engineering plays a vital role in enhancing the predictive performance of LSTM models by incorporating temporal dependencies and contextual cues. This process was performed separately for temperature and precipitation datasets, taking into account the distinct characteristics of each variable, particularly in how lag features were selected.

For temperature forecasting, False Nearest Neighbors (FNN) analysis was employed to identify an appropriate input sequence length and lag structure. The FNN results, which provided an estimate of the optimal embedding dimension, guided the creation of lagged features that capture temporal dependencies effectively. Specifically, past temperature values at 24, 48, and 72-hour intervals were selected, as these aligned with observed daily and multi-day cycles from ACF and PACF plots. Additionally, 24-hour rolling statistics—namely the rolling mean and standard deviation—were computed to summarize short-term temperature trends and volatility. To further enrich the temporal context, time-based features such as hour of the day, day of the week, day of the year, and month were derived from the datetime index, allowing the model to leverage cyclical and seasonal patterns.

In contrast, for precipitation forecasting, FNN analysis did not yield a clear embedding dimension, likely due to the sporadic and zero-inflated nature of rainfall data. As a result, lagged features were chosen empirically based on domain knowledge and insights from ACF/PACF plots. Lag intervals of 1, 3, 6, 12, 24, and 72 hours were selected to capture both short-term rainfall dynamics and historical behavior. Similar to the temperature dataset, rolling statistics over a 24-hour window were included to account for accumulated rainfall and variability. The same time-based features extracted for temperature were also applied here to help model periodic rainfall tendencies.

By customizing the feature engineering process to reflect the temporal structure and statistical properties of each variable, the LSTM models were equipped with richer inputs, enabling them to learn more effectively from the available data.

4.6 Data Normalization

LSTM models require inputs on similar scales to stabilize and speed up learning. The MinMaxScaler from sklearn.preprocessing was used for normalization, scaling all features including the target to a [0, 1] range. To prevent data leakage, the scaler was fitted exclusively on the training data before being applied to transform both the training and validation sets. This same fitted scaler was subsequently used to inverse-transform model predictions back to their original measurement scales

(degrees Celsius for temperature and millimeters for precipitation). For consistent preprocessing during future inference, the scaler object was preserved using pickle serialization.

4.7 Sequence Generation for LSTM

The data was restructured into the 3D input format required by LSTM models, organized as [samples, timesteps, features] through a custom sliding window implementation. Input sequences (X) captured the past 24 hours of observations, while output sequences (y) varied based on forecasting type: for single-step prediction, the target was the next 1-hour value, whereas multi-step forecasting directly predicted the subsequent 24-hour values in a single output. This framework was uniformly applied to both temperature and precipitation models, though precipitation models employed empirically determined sequence lengths to accommodate their more irregular patterns.

4.8 Time-Based Data Splitting

The dataset was partitioned chronologically to preserve temporal relationships, with the earliest 80% of time steps allocated to training and the most recent 20% reserved for validation. This approach intentionally avoided random shuffling to maintain the natural sequence of observations, thereby creating a validation set that genuinely represents unseen future conditions. Such temporal splitting provides a realistic evaluation framework by testing the model's ability to generalize to future time periods beyond its training horizon.

5 Methodology

This section outlines the complete methodology adopted for developing LSTM-based models for forecasting temperature and precipitation using univariate time series data from the Vaitarna basin. The methodology includes the design and experimentation of multiple LSTM architectures, preparation of input-output sequences for both single-step and multi-step forecasting, and consistent hyperparameter tuning. Given the distinct nature of the two variables—temperature being smooth and seasonal, and precipitation being highly volatile—separate models were developed for each, following a parallel but independent pipeline.

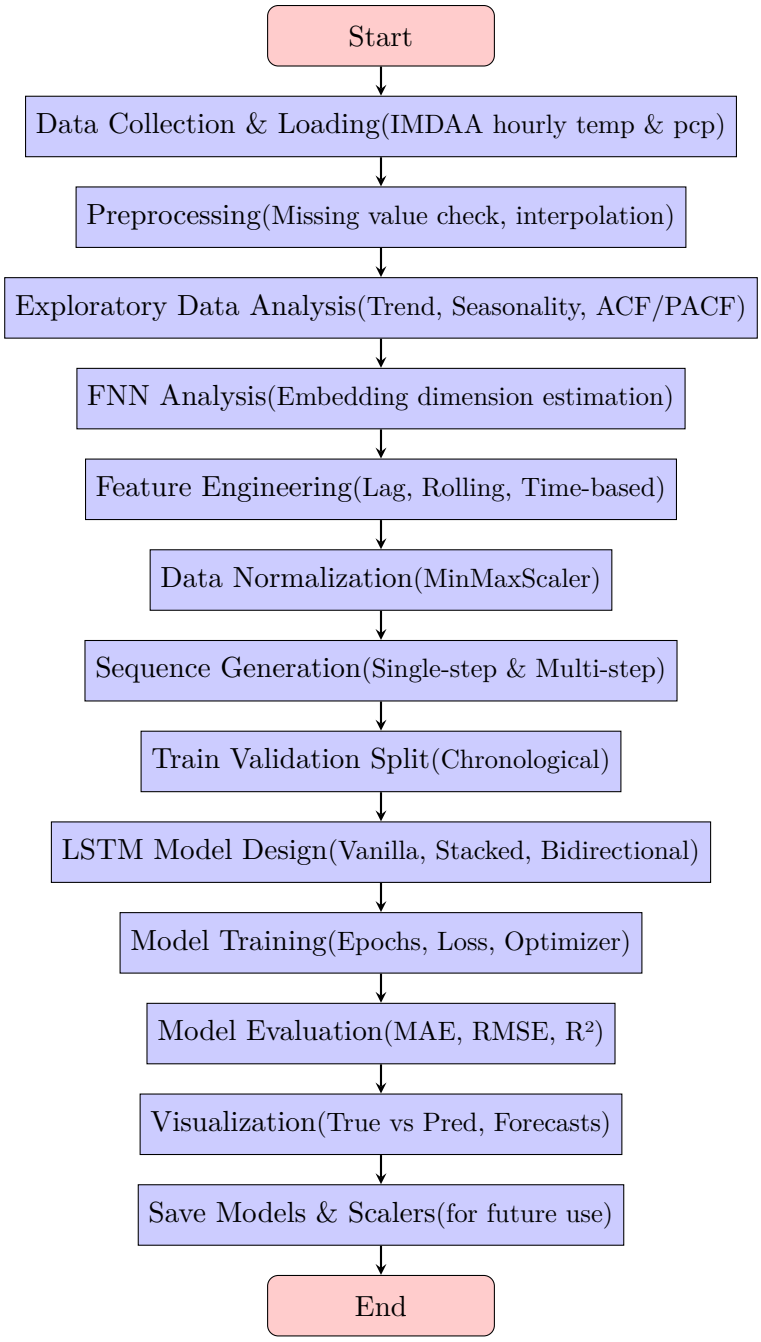


Figure 3: Complete LSTM forecasting methodology workflow showing the sequence from data collection through model development to final outputs. The process emphasizes thorough data preparation, feature engineering, and model evaluation steps characteristic of time series forecasting.

5.1 LSTM Model Development

To effectively model temporal dependencies and capture both short-term and seasonal patterns in the data, three widely used LSTM architectures were implemented and evaluated: Vanilla LSTM, Stacked LSTM, and Bidirectional LSTM. Each architecture was applied separately to the single-step (predicting the next hour) and multi-step (predicting the next 24 hours) forecasting tasks.

The Vanilla LSTM model serves as the baseline architecture. It consists of a single LSTM layer followed by a Dense layer. For single-step forecasting, the LSTM layer uses return_sequences=False, and the output is passed to a Dense layer with one neuron to predict the next value. For multi-step forecasting, the same LSTM structure is retained, but the Dense layer contains 24 neurons to predict a sequence of the next 24 hourly values in one shot. This direct multi-output strategy simplifies the decoding process and allows for efficient training.

The Stacked LSTM architecture enhances the learning capacity of the network by stacking multiple LSTM layers. Intermediate LSTM layers are configured with return_sequences=True to pass their full output sequences to the subsequent layers. The final LSTM layer uses return_sequences=False, followed by a Dense layer, similar to the Vanilla model, for output generation. This deeper architecture allows the model to learn more abstract temporal patterns by capturing hierarchical representations of the input sequence.

The Bidirectional LSTM architecture is designed to process input sequences in both forward and backward directions. In this setup, an LSTM layer is wrapped within a Bidirectional layer, allowing the network to extract information from both ends of the input sequence simultaneously. Although in forecasting tasks future time steps beyond the current window are unknown, the bidirectional mechanism still improves the model's ability to interpret the structure of the input sequence more comprehensively. This architecture proved particularly beneficial for temperature, where strong periodicity exists within the input window.

All architectures were implemented using the Keras Sequential API, and consistent design patterns were followed to maintain comparability across different models.

5.2 Input-Output Sequence Setup

The LSTM models were designed to handle supervised input-output pairs derived from historical sequences using a sliding window approach. For both temperature and precipitation, input sequences were constructed using a fixed look-back window, typically 24 hours in length. Each sequence included not only the primary variable (temperature or precipitation) but also a set of engineered features such as lagged values, rolling statistics, and time-based indicators.

For single-step forecasting, the target output was the value at the immediate next time step. Thus, the output vector had a shape of (samples, 1). For multi-step forecasting, the model was trained to predict the next 24 values in sequence, giving an output shape of (samples, 24). This direct strategy was preferred over recursive methods to avoid error accumulation over the forecast horizon.

The input data shape for all models conformed to the standard LSTM format:

(samples, timesteps, features). Here, timesteps corresponded to the input sequence length (typically 24), and features represented the number of variables provided to the model at each time step.

5.3 Hyperparameter Configuration

In order to ensure fair comparison across different LSTM architectures and forecasting tasks, a consistent set of training hyperparameters was adopted for both temperature and precipitation models. All models were compiled using the Adam optimizer, known for its adaptive learning rate and efficient convergence. The Mean Squared Error (MSE) was used as the loss function, given its suitability for continuous-valued regression problems.

The models were trained for 10 epochs, which was empirically found to balance training duration and convergence. A batch size of 64 was used to manage memory efficiently and ensure stable gradient updates. No shuffling was applied to the input data, as preserving the temporal order is critical in time series forecasting.

5.4 Separate Modeling for Temperature and Precipitation

Although the overall modeling approach was shared, separate LSTM models were developed for temperature and precipitation due to their differing statistical properties and temporal behaviors.

Temperature typically exhibits smooth, gradual trends and well-defined seasonal cycles. Therefore, the models could effectively leverage periodic lagged features and benefited significantly from the use of Bidirectional LSTMs, which captured dependencies within the input sequence more comprehensively.

In contrast, precipitation data is inherently more erratic and sparse, with abrupt spikes and prolonged dry periods. Consequently, the same architectures were applied to precipitation using empirically chosen lag features (since FNN analysis did not yield a clear embedding dimension). Although the underlying architectures remained identical, each model was trained on its respective dataset independently, and the weights, scalers, and forecasts were managed separately.

This separation ensured that each model was tailored to the unique temporal dynamics of its variable, ultimately improving forecasting accuracy and interpretability.

6 Experiments and Results

This section presents and analyzes the experimental results obtained from training and evaluating different LSTM architectures, Vanilla, Stacked, and Bidirectional,

for both temperature and precipitation forecasting tasks. Each model was evaluated for both single-step (1-hour ahead) and multi-step (24-hour ahead) forecasting, and assessed using standard regression metrics: MAE (Mean Absolute Error), RMSE (Root Mean Squared Error), and R^2 (Coefficient of Determination). The experiments also included visual comparisons between predicted and true values, and error analysis based on both qualitative plots and quantitative results.

6.1 Training and Validation Loss Analysis

This section analyzes the training and validation loss curves for all LSTM model variants (Vanilla, Stacked, and Bidirectional) across temperature and precipitation forecasting tasks. Each architecture was evaluated for both single-step and multi-step (24-hour) horizons using Mean Squared Error (MSE) loss.

Temperature Forecasting

For single-step forecasting, all three models demonstrated good convergence. The Vanilla and Bidirectional LSTMs showed smooth and closely aligned loss curves, with low final validation losses ($\sim$3–4$\times10^{-5}$), indicating strong generalization. The Stacked LSTM showed slightly higher variance in validation loss, with signs of minor overfitting.

In multi-step forecasting, the Vanilla LSTM outperformed others, achieving the lowest final losses (train: $\sim$2.0$\times10^{-4}$, val: $\sim$2.5$\times10^{-4}$). It captured long-range temporal patterns effectively. Stacked and Bidirectional LSTMs also converged well but with slightly higher validation error and slower convergence.

Precipitation Forecasting

Precipitation models showed more instability due to the irregular nature of rainfall. In single-step tasks, training losses decreased steadily, but validation losses remained higher ($\sim$7–8$\times10^{-5}$), especially in Bidirectional model, suggesting overfitting.

In the multi-step precipitation forecasting task, all three LSTM variants showed signs of overfitting. The Vanilla LSTM maintained a relatively stable validation loss ($\sim$1.2$\times10^{-4}$), but the training loss continued to decline, indicating a widening gap and limited generalization. The Stacked LSTM overfit more significantly, with a persistent 3$\times$ gap between training and validation losses despite continued training improvement. The Bidirectional LSTM exhibited early plateauing in both losses, with the validation loss remaining consistently higher, suggesting rapid convergence but poor generalization beyond the second epoch.

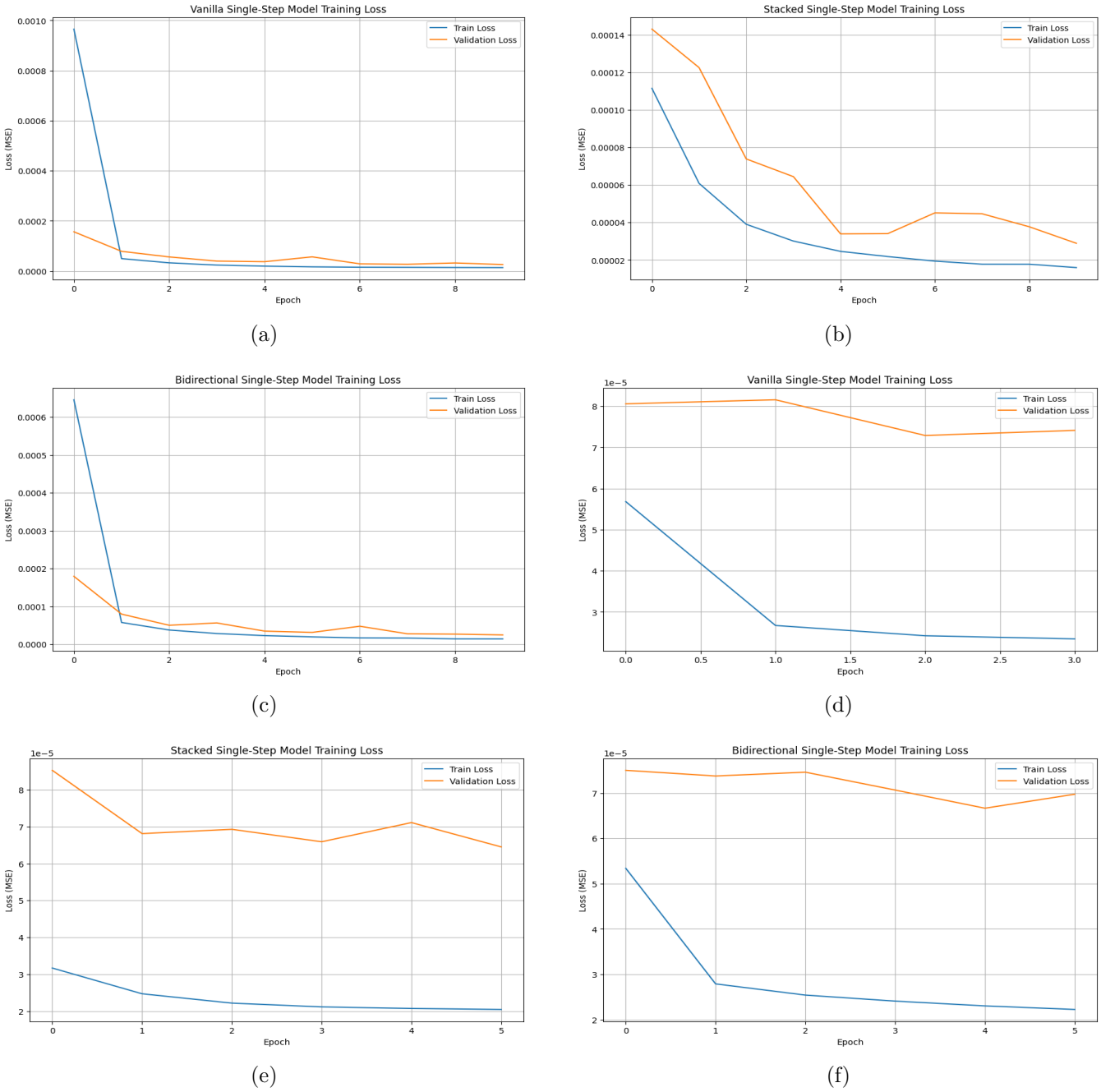
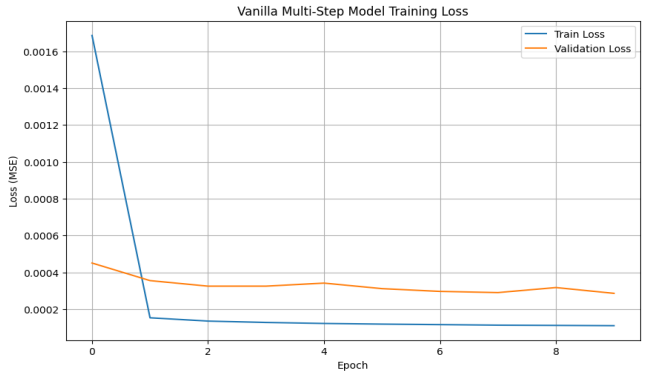
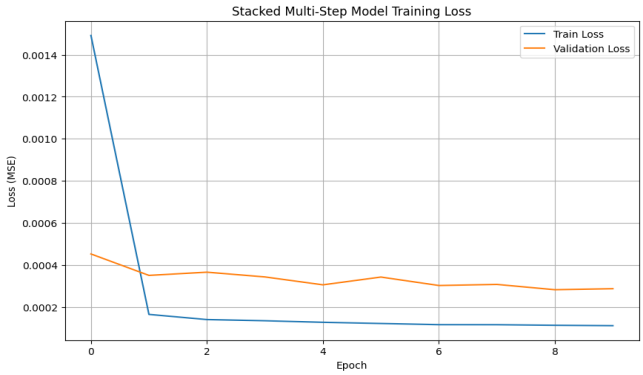


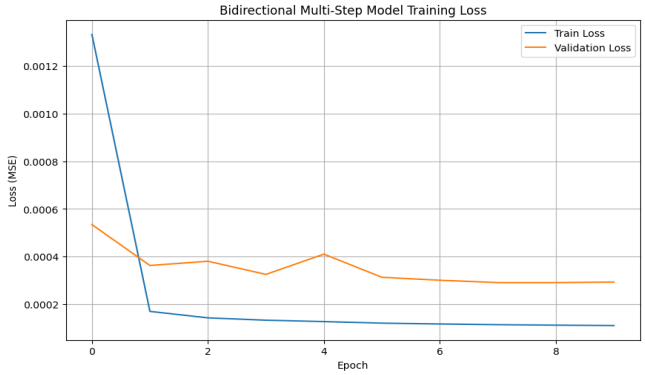
Figure 4: Training loss curves showing (a) Vanilla LSTM, (b) Stacked LSTM, and (c) Bidirectional LSTM performance on single-step temperature forecasting, along with (d) Vanilla LSTM, (e) Stacked LSTM, and (f) Bidirectional LSTM results for single-step precipitation prediction. Temperature models exhibit smoother convergence compared to precipitation models due to their more periodic nature.



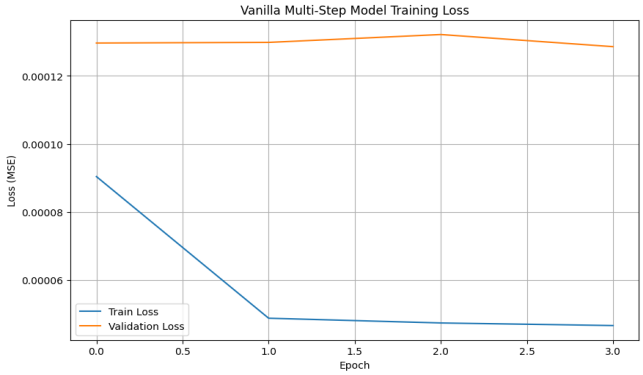
(g)



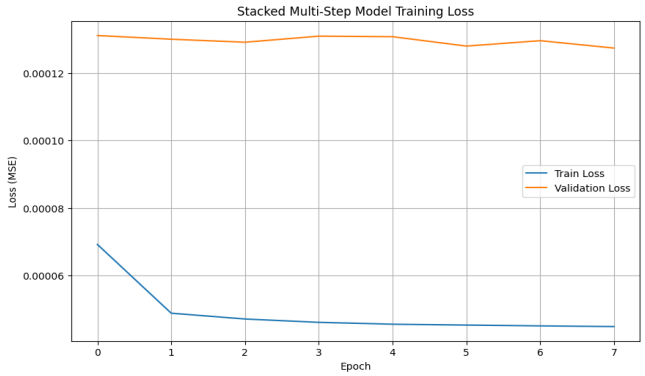
(h)



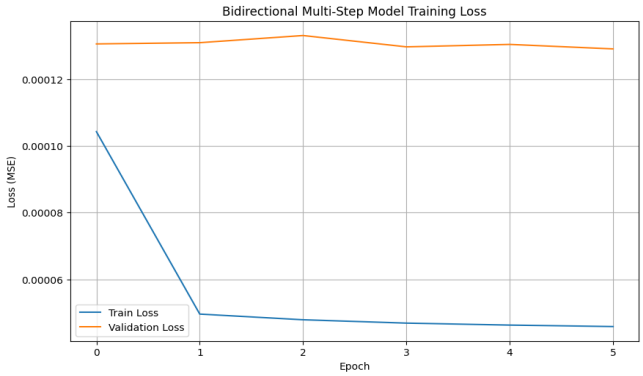
(i)



(j)



(k)



(l)

Figure 4: Continued: Multi-step forecasting results showing (g) Vanilla LSTM, (h) Stacked LSTM, and (i) Bidirectional LSTM performance on temperature prediction, along with (j) Vanilla LSTM, (k) Stacked LSTM, and (l) Bidirectional LSTM results for precipitation. The increased loss values in multi-step predictions reflect the compounding of errors over longer forecasting horizons, particularly noticeable in the more volatile precipitation models.

6.2 Evaluation Metrics: MAE, RMSE, and R^2

Quantitative performance was evaluated using three standard regression metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R^2 (Coefficient of Determination). MAE represents the average magnitude of the prediction

errors, providing a straightforward measure of overall accuracy. RMSE, by squaring the errors before averaging, places greater emphasis on larger errors, making it useful for highlighting occasional but significant deviations. R^2 indicates how well the predicted values explain the variance in the actual values, with higher values signifying better model fit and predictive capability.

Table 2: Temperature Forecasting Results

Model	**Task**	**MAE (°C)**	**RMSE (°C)**	**R^2**
Bidirectional	Single-Step	0.0181	0.0431	0.9994
Vanilla	Single-Step	0.0172	0.0428	0.9994
Stacked	Single-Step	0.0229	0.0456	0.9993
Vanilla	Multi-Step	0.0869	0.1433	0.9927
Stacked	Multi-Step	0.0872	0.1435	0.9927
Bidirectional	Multi-Step	0.1044	0.1545	0.9916

Bidirectional LSTM performed best for single-step forecasting, achieving the lowest MAE and RMSE. For multi-step forecasting, Vanilla and Stacked LSTM models showed slightly better overall performance compared to Bidirectional. All models achieved R^2 values greater than 0.99, indicating excellent capability in modeling temperature dynamics.

Table 3: Precipitation Forecasting Results

Model	**Task**	**MAE (mm)**	**RMSE (mm)**	**R^2**
Stacked LSTM	Single-Step	0.0949	0.5075	0.6774
Bidirectional LSTM	Single-Step	0.0988	0.5257	0.6539
Vanilla LSTM	Single-Step	0.2422	0.5432	0.6304
Stacked LSTM	Multi-Step	0.1634	0.7085	0.3714
Vanilla LSTM	Multi-Step	0.1841	0.7097	0.3692
Bidirectional LSTM	Multi-Step	0.1788	0.7087	0.3710

Stacked LSTM outperformed other models for precipitation forecasting in both single-step and multi-step tasks. Bidirectional models were competitive but slightly less accurate. The models achieved moderate R^2 values, ranging from 0.63 to 0.68 for single-step forecasting, and around 0.37 for multi-step forecasting. This indicates that while the models were able to capture short-term rainfall patterns reasonably well, their performance declined for extended forecasts due to the irregular and sparse nature of precipitation data.

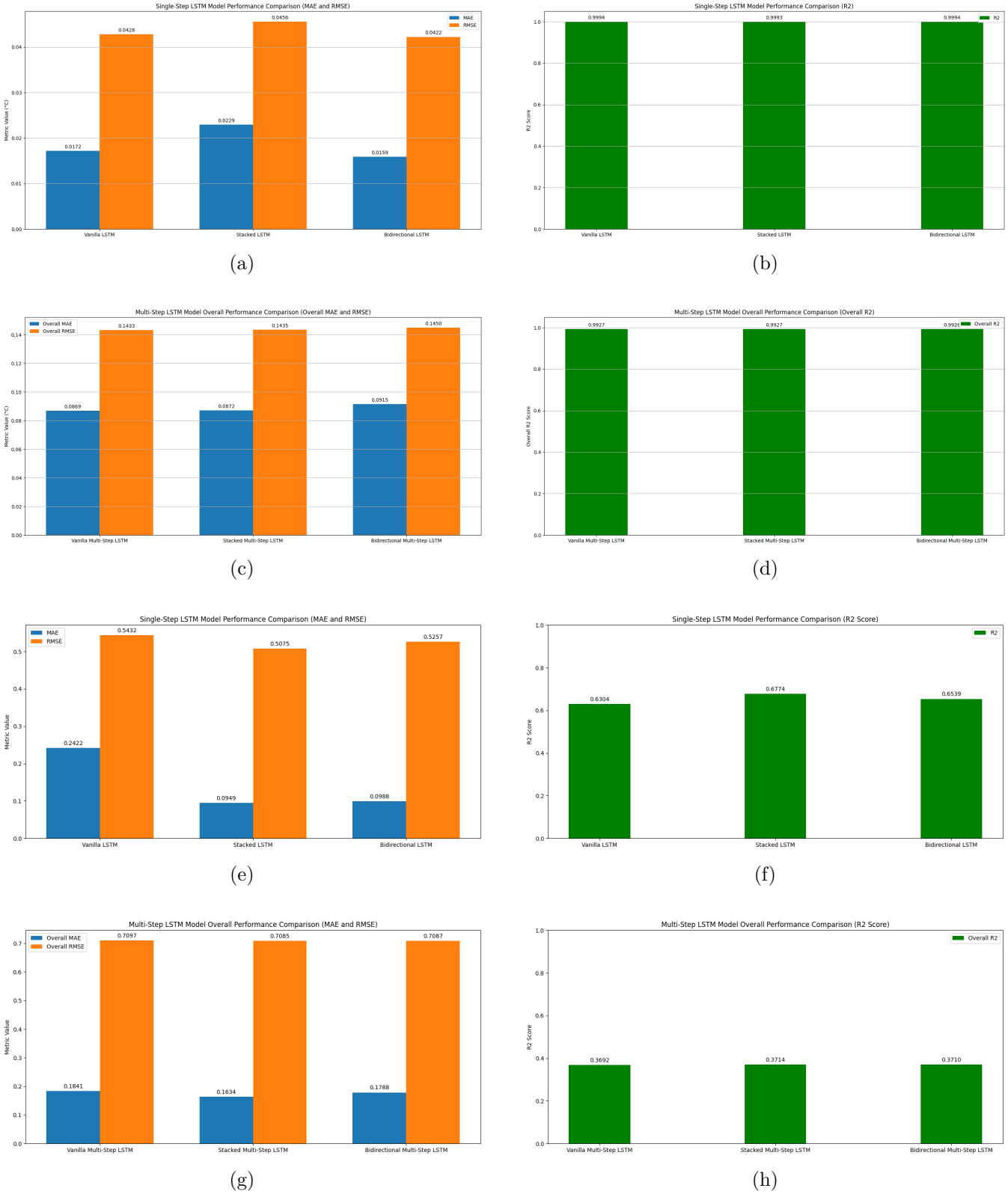


Figure 5: Error analysis and model performance across forecasting tasks. Plots (a) and (b) show single-step temperature forecasting results, with (a) displaying error distributions and (b) presenting R^2 values. Plots (c) and (d) correspond to multi-step temperature forecasting, showing error patterns and goodness-of-fit respectively. Precipitation forecasting performance is shown in (e)-(h), where (e) illustrates single-step error distributions, (f) shows single-step R^2 values , (g) displays multi-step errors , and (h) presents multi-step R^2 metrics . The temperature models generally exhibit tighter error distributions and higher R^2 values compared to precipitation models, particularly for multi-step forecasting where precipitation errors show greater variance due to rainfall's stochastic nature.

6.3 Comparison of LSTM Variants

For temperature forecasting, the Bidirectional LSTM emerged as the top performer in the single-step prediction task, owing to its ability to effectively learn bidirectional temporal context within the input window. However, in the multi-step forecasting scenario, the Vanilla and Stacked LSTM models achieved slightly lower prediction errors compared to the Bidirectional model, indicating that increased architectural complexity does not necessarily translate to improved long-range forecasting accuracy.

In the case of precipitation forecasting, the Stacked LSTM model demonstrated a clear advantage over both Vanilla and Bidirectional models. Its deeper architecture was better suited for capturing the irregular and zero-inflated patterns typical of rainfall data. While Bidirectional LSTMs are effective for structured and smooth time series, they provided limited improvement for precipitation due to the lack of coherent temporal continuity in such data.

A key observation from this comparative analysis is that the performance of LSTM variants is highly dependent on the characteristics of the target variable. Smooth and continuous variables like temperature benefit from context-aware architectures such as the Bidirectional LSTM, whereas discontinuous or spiky variables like rainfall respond better to deeper models like the Stacked LSTM, which are capable of learning complex hierarchical representations.

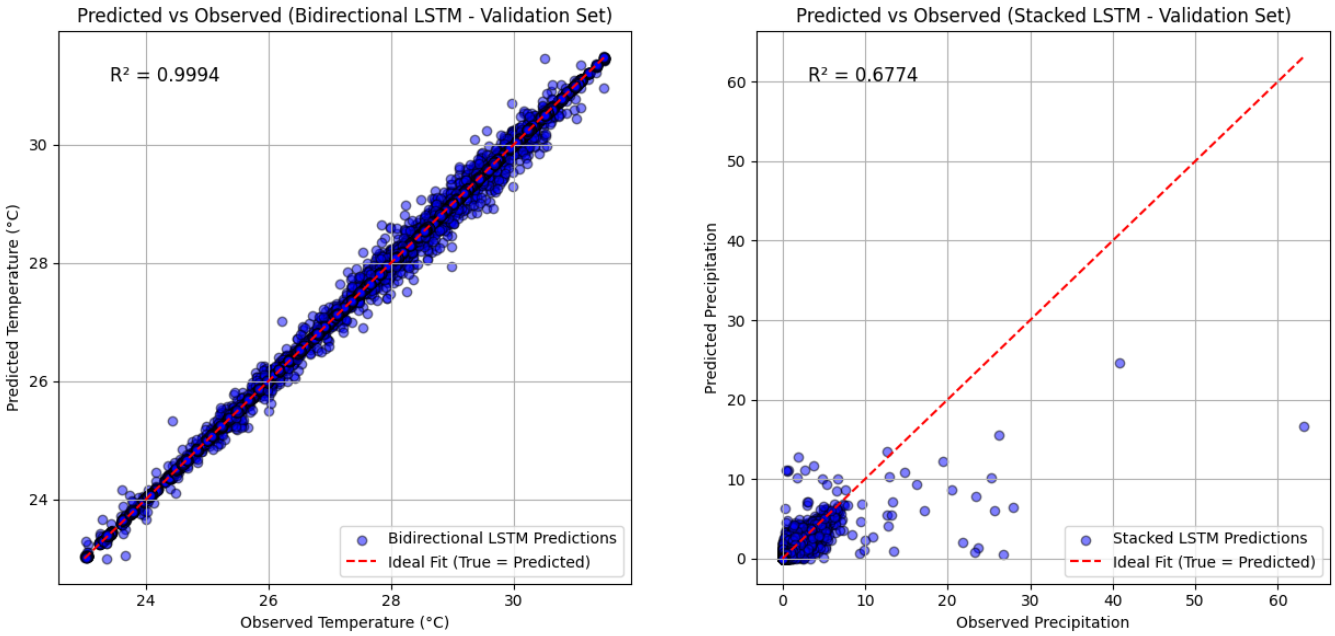
6.4 Forecast Visualization: True vs Predicted

Visualization of model predictions provides a qualitative assessment of performance and complements the evaluation metrics.

Single-Step Forecast Plots

In the case of single-step forecasting, the predicted temperature values closely aligned with the ground truth across all models, indicating strong short-term predictive capability. Among them, the Bidirectional LSTM demonstrated minimal lag and was particularly effective at tracking the peaks and troughs in temperature variation, further validating its ability to capture temporal dependencies. For precipitation, however, the Stacked LSTM was the only model that consistently captured both the occurrence and absence of rainfall, highlighting its superior handling of the irregular and sparse nature of precipitation data.

Forecast Accuracy Visualization



(a) (b)

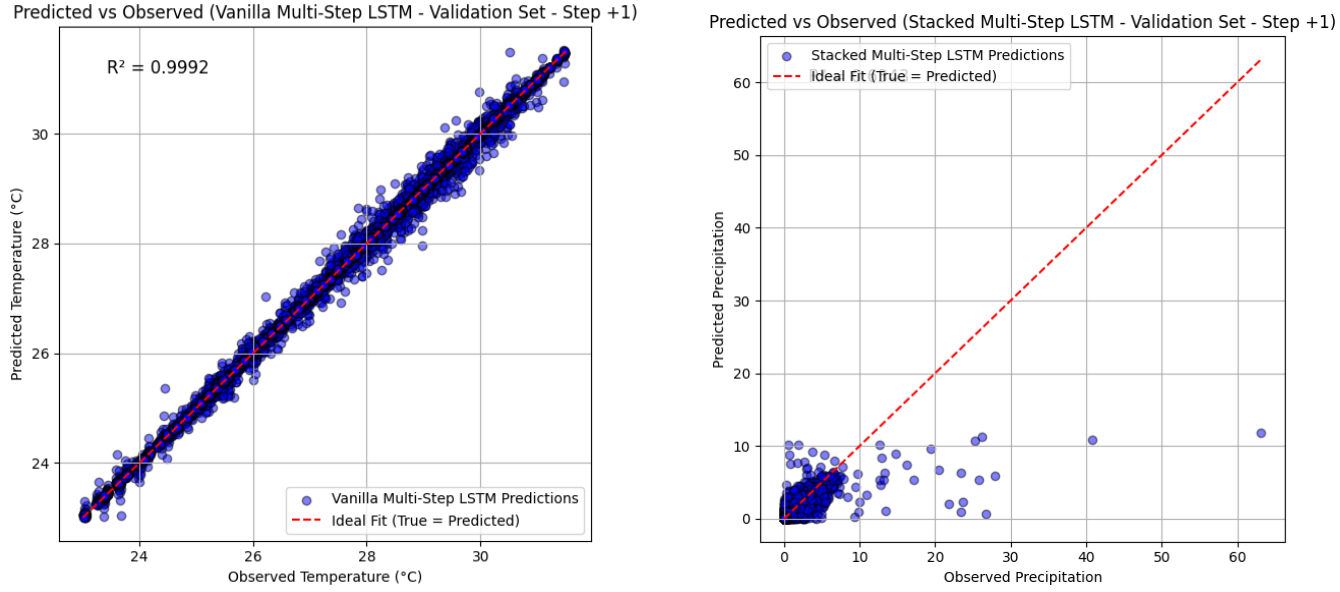
Figure 6: Observed vs predicted values for single-step forecasting of (a) temperature and (b) precipitation.

In the temperature plot, the points are tightly clustered along the 1:1 dashed line, indicating highly accurate predictions with an R^2 value of 0.999 and a low MAE of 0.0181°C. In contrast, the precipitation plot shows greater dispersion around the 1:1 line, reflecting the higher variability and difficulty in predicting rainfall, with an R^2 value of 0.677 and an MAE of 0.0949 mm.

Multi-Step Forecast Plots

The multi-step forecast plots, covering a 24-hour horizon, reveal that all models deliver their most accurate predictions during the early forecast steps—approximately the first one to six hours—while prediction error progressively increases toward the final lead time at +24 hours. Overall, the predicted and observed curves follow similar trends, though discrepancies in magnitude emerge during periods of abrupt change. For temperature, the Vanilla and Stacked LSTM variants preserve notably smoother trajectories across the full 24-hour window, demonstrating a consistent ability to track the underlying pattern even as forecast uncertainty grows with time.

Multi-step Forecast Accuracy



(a) (b)

Figure 7: Observed vs predicted values for 24-hour forecasting of (a) temperature and (b) precipitation. The diagonal dashed line indicates perfect predictions. Note the different scales reflecting each variable's characteristics.

For temperature, the multi-step scatter plot (a) shows a high correlation between predicted and observed values, with an R^2 score of 0.992 and a low MAE of 0.0869°C. The predictions closely align with the ground truth, though there is a tendency to slightly underpredict during extreme temperature values. In contrast, the precipitation scatter plot (b) demonstrates greater spread, with a lower R^2 score of 0.371 and a higher MAE of 0.1634mm. This reflects the inherent difficulty in modeling sudden and sparse rainfall events, which the model often struggles to predict with high precision.

Future Forecast Plot

Forecasts immediately following the validation window were generated and visualized for both temperature and precipitation time series. For temperature, the Bidirectional LSTM effectively extended recent trends in a plausible and consistent manner, capturing the smooth temporal progression. In the case of precipitation, the Stacked LSTM demonstrated good performance in predicting dry periods accurately; however, it occasionally underestimated brief and sudden rainfall events, highlighting the inherent difficulty in modeling sharp, irregular spikes in precipitation data.

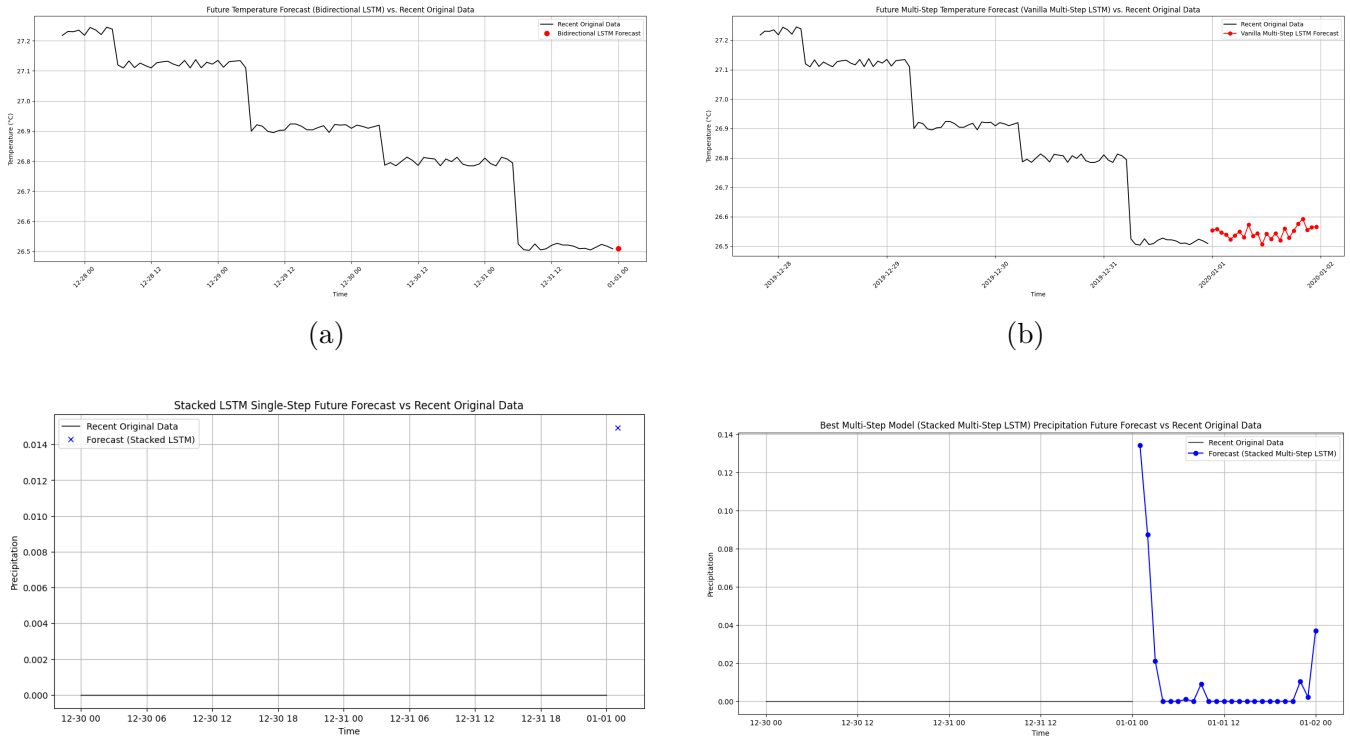


Figure 8: presents forecast results for both temperature and precipitation. Subfigure (a) shows the single-step temperature forecast using the Bidirectional LSTM, while (b) illustrates the 24-hour multi-step forecast using the Vanilla LSTM. Subfigure (c) displays the single-step precipitation forecast generated by the Stacked LSTM, and (d) presents the corresponding 24-hour multi-step forecast.

6.5 Error Analysis

The magnitude of forecasting errors reveals important insights into model behavior. In all cases, the Root Mean Squared Error (RMSE) was higher than the Mean Absolute Error (MAE), indicating the presence of occasional large errors that disproportionately influenced the RMSE values. These significant deviations were most frequently observed during abrupt weather transitions, such as sharp temperature drops or sudden rainfall events.

In multi-step forecasting, a clear pattern of error progression was evident. Errors were smallest during the initial forecast steps and increased steadily as the prediction horizon extended. This behavior aligns with a known limitation of direct multi-step forecasting, where uncertainties accumulate over longer time horizons, reducing predictive accuracy.

Each LSTM architecture exhibited distinct error tendencies. The Bidirectional LSTM often smoothed out rapid transitions, leading to underestimation of extreme values. The Stacked LSTM, by contrast, demonstrated improved sensitivity to

precipitation spikes, though it occasionally overreacted to random fluctuations or noise. Meanwhile, the Vanilla LSTM generally offered stable performance but was less responsive to sudden changes in weather trends. These differences highlight the importance of aligning model architecture with the dynamic characteristics of the target variable.

7 Conclusion and Future Work

This project successfully explored and evaluated deep learning techniques, specifically Long Short-Term Memory (LSTM) networks, for time series forecasting of temperature and precipitation using hourly data from the Indian Monsoon Data Assimilation and Analysis reanalysis (IMDAA) for the Vaitarna basin. Three LSTM architectures, Vanilla, Stacked, and Bidirectional, were implemented and tested across both single-step and multi-step (24-hour) forecasting horizons.

The temperature forecasting models demonstrated excellent performance across all configurations. The Bidirectional LSTM outperformed other models in the single-step temperature forecasting task, while the Vanilla LSTM delivered the best performance for multi-step forecasting, achieving the lowest overall validation error and strong generalization. All temperature models achieved very low Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), with R^2 values exceeding 0.99, indicating highly accurate predictions and excellent generalization.

In contrast, precipitation forecasting posed significant challenges due to the sparse, non-periodic, and irregular nature of rainfall data. While the single-step models achieved moderate R^2 scores between 0.63 and 0.68, the multi-step models struggled with long-range prediction, yielding lower R^2 values around 0.37. All three precipitation models exhibited signs of overfitting in the multi-step task, with validation losses plateauing or diverging from training losses. This reflects the difficulty in modeling stochastic rainfall behavior with univariate sequence data alone.

7.1 Limitations

- The models were trained solely on univariate time series data, limiting their ability to leverage external meteorological variables (e.g., humidity, pressure, wind).
- Spatial context was not incorporated, despite the dataset covering multiple grid points across the basin.
- Standard LSTM architectures, although effective for temperature, struggled to generalize for precipitation, especially over longer horizons.

7.2 Future Work

To overcome these challenges and enhance forecasting accuracy, the following directions are proposed:

- Hybrid modeling approaches that combine LSTM with statistical models (e.g., ARIMA) or symbolic methods (e.g., Genetic Programming) may improve generalization on sparse or noisy datasets.
- Multivariate LSTM models could be developed by including additional environmental variables such as pressure, wind speed, humidity, and soil moisture to enrich the feature space.
- Spatial modeling techniques, such as ConvLSTM or attention-based models, could capture dependencies across nearby locations, improving regional rainfall prediction.
- Further hyperparameter tuning, longer training, and the use of regularization techniques (dropout, early stopping) can help mitigate overfitting, especially in complex architectures like Stacked and Bidirectional LSTMs.

References

Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: Neural computation 9.8 (1997), pp. 1735–1780.

Frederik Kratzert et al. "Rainfall–runoff modelling using long short-term memory (LSTM) networks". In: Hydrology and Earth System Sciences 22.11 (2018), pp. 6005–6022.

Bibhuti Bhusan Sahoo et al. "Long short-term memory (LSTM) recurrent neural network for low-flow hydrological time series forecasting". In: Acta Geophysica 67.5 (2019), pp. 1471–1481.

Zhice Fang et al. "Predicting flood susceptibility using LSTM neural networks". In: Journal of Hydrology 594 (2021), p. 125734.

Binglin Li et al. "An Empirical Modal Decomposition-Improved Whale Optimization Algorithm-Long Short-Term Memory Hybrid Model for Monitoring and Predicting Water Quality Parameters". In: Sustainability 15.24 (2023), p. 16816.

Hyeontae Moon, Sunkwon Yoon, and Youngil Moon. "Urban flood forecasting using a hybrid modeling approach based on a deep learning technique". In: Journal of Hydroinformatics 25.2 (2023), pp. 593–610.

Xin Xiang et al. "An explainable deep learning model based on hydrological principles for flood simulation and forecasting". In: EGUsphere 2025 (2025), pp. 1–37.

Fabian Pedregosa et al. "Scikit-learn: Machine learning in Python". In: the Journal of machine Learning res 12 (2011), pp. 2825–2830.

Francois Chollet et al. "Keras: Deep learning for humans". In: GitHub. Inc (2015).

Martín Abadi et al. "{TensorFlow}: a system for {Large-Scale} machine learning". In: 12th USENIX symposium on operating systems design and implementation (OSDI 16). 2016, pp. 265–283.

Raghavendra Ashrit et al. "IMDAA regional reanalysis: Performance evaluation during Indian summer monsoon season". In: Journal of Geophysical Research: Atmospheres 125.2 (2020), e2019JD030973.

Yanhong Dou et al. "A framework for merging precipitation retrievals and gauge-based observations based on a novel concept namely virtual gauges". In: Journal of Hydrology 620 (2023), p. 129506. ISSN: 0022-1694. DOI: 10.1016/j.jhydrol.2023.129506. URL: https://www.sciencedirect.com/science/article/pii/S0022169423004481.