Analysis of Rainfall Trends with Satellite Data

Master of Technology
in
Computer Science and Engineering

by

Prakhar Prajapati(25CS60R66)
Ansh Gupta (25CS60R34)

Supervisor

Dr. SK Ghosh & Dr. S Sural

Indian Institute of Technology Kharagpur
India

November-2025

Contents

Abstract

THIS investigates rainfall variability and prediction in India's Delhi-NCR region, a critical component to strengthening urban resilience, optimizing agricultural planning and improving climate risk management. The project leverages high-resolution, satellite-derived rainfall estimates from NASA's Global Precipitation Measurement (GPM) Integrated Multi-satellitE Retrievals for GPM (IMERG) dataset [3], spanning two decades (2004–2024), to detect long-term temporal trends and forecast future rainfall patterns. The project employs a comprehensive methodological framework combining statistical trend tests, variability analysis, and advanced machine learning predictive modeling. The non-parametric Mann–Kendall test, supplemented with Sen's slope estimator, reveals a statistically significant increasing rainfall trend in January (+1.69 mm/year, $p = 0.015$), indicating potential shifts in winter precipitation dynamics. In contrast, the core South Asian monsoon months (June–September) exhibit overall stability with no significant trends detected in seasonal totals. For predictive modeling, the project implements and compares three machine learning algorithms: Random Forest (RF) [4], eXtreme Gradient Boosting (XGBoost) [5], and Long Short-Term Memory (LSTM) [6] neural networks. Among these, LSTM demonstrates superior forecasting capability, achieving a root mean squared error (RMSE) of 31.95 mm and a coefficient of determination (R^2) of 0.859 on validation data, attributed to its effectiveness in capturing sequential temporal dependencies. The ensemble rainfall forecasts generated by the project indicate relatively stable monsoon rainfall in the near future but reveal substantial uncertainty during winter (January–February) and post-monsoon (October–December) periods, posing challenges for seasonal water resource planning. This project demonstrates the significant potential of integrating satellite-based precipitation observations with advanced machine learning techniques for robust, high-resolution rainfall trend analysis and prediction. The findings underscore the value of such data-driven approaches for developing adaptive strategies in urban water resource management, disaster preparedness, and sustainable agriculture in rapidly urbanizing regions like Delhi-NCR.

1 Introduction

Rainfall is one of the most significant meteorological parameters influencing agriculture, hydrology, and urban sustainability. In the context of rapid urbanization and climate variability, regions like Delhi NCR face increasing challenges related to water management, drought, and flooding. Understanding historical rainfall trends and accurately predicting future rainfall are essential for informed decision-making in agriculture, infrastructure planning, and disaster mitigation.

This project leverages satellite-based rainfall observations from NASA's GPM IMERG Monthly dataset to perform long-term rainfall analysis. The dataset provides a high-resolution ($0.1^\circ \times 0.1^\circ$) global rainfall product derived from multiple satellite sensors, ensuring data reliability and spatial coverage. The study integrates remote sensing, statistical analysis, and machine learning to produce an end-to-end rainfall analysis and forecasting pipeline.

2 Methodology

This section describes the complete end-to-end methodology followed for rainfall trend analysis and forecasting in the Delhi NCR region. The workflow integrates data retrieval, preprocessing, exploratory visualization, trend detection, feature engineering, machine learning model training, and future rainfall prediction.

2.1 Data Acquisition using Google Earth Engine

Rainfall data was sourced from the NASA Global Precipitation Measurement (GPM) IMERG Monthly V07 dataset, accessed via the Google Earth Engine (GEE) platform. GEE provides cloud-based access to large-scale satellite archives and allows spatial filtering and aggregation through efficient APIs.

A bounding box for Delhi NCR (76.5°E–78.5°E, 28.0°N–29.5°N). The IMERG dataset contains the variable precipitation, representing mean rainfall in mm/hour.

To convert this into monthly rainfall (mm/month), each monthly image was multiplied by the number of hours in that month:

$$\text{Rainfall}_{\text{mm/month}} = \text{Rainfall}_{\text{mm/hr}} \times (\text{days in month} \times 24)$$

This conversion ensures physical interpretability aligned with climatological conventions. The results were stored as a structured DataFrame containing columns for date, year, month, and computed rainfall metrics.

Caching using Pickle files was employed to avoid redundant Earth Engine calls and improve reproducibility.

2.2 Data Preprocessing and Organization

After data extraction, rainfall values were aggregated and summarized for descriptive statistics. Each record was tagged with `month_name` using Python's `calendar` library for readability and grouped using Pandas operations.

A series of integrity checks were performed:

- Missing value detection (`isna()` check)
- Range validation for extreme outliers
- Consistency check between total record count and expected monthly intervals (2004–2024)

Two directories—`data/` and `results/`—were automatically created to store processed datasets and generated plots. The dataset was serialized both as `.csv` and `.pkl` for use in downstream ML stages. [7]

2.3 Exploratory Data Analysis (EDA)

Exploratory analysis was performed to visually and statistically understand rainfall dynamics using Matplotlib and Seaborn.

- **Monthly Time Series:** Plotted to observe inter-annual and seasonal variability across two decades. This reveals periodic monsoon peaks and low rainfall months.
- **Monthly Climatology:** Average rainfall by month (Jan–Dec) was computed to understand intra-annual seasonality.
- **Annual Aggregation:** Yearly sums highlight long-term variability, serving as input for trend testing.
- **Rainfall Distribution:** Histogram of monthly rainfall illustrates skewness and heavy-tail behavior common in rainfall datasets.

EDA outputs were saved `.png` figures for reporting (`01_exploratory_analysis.png`). [7]

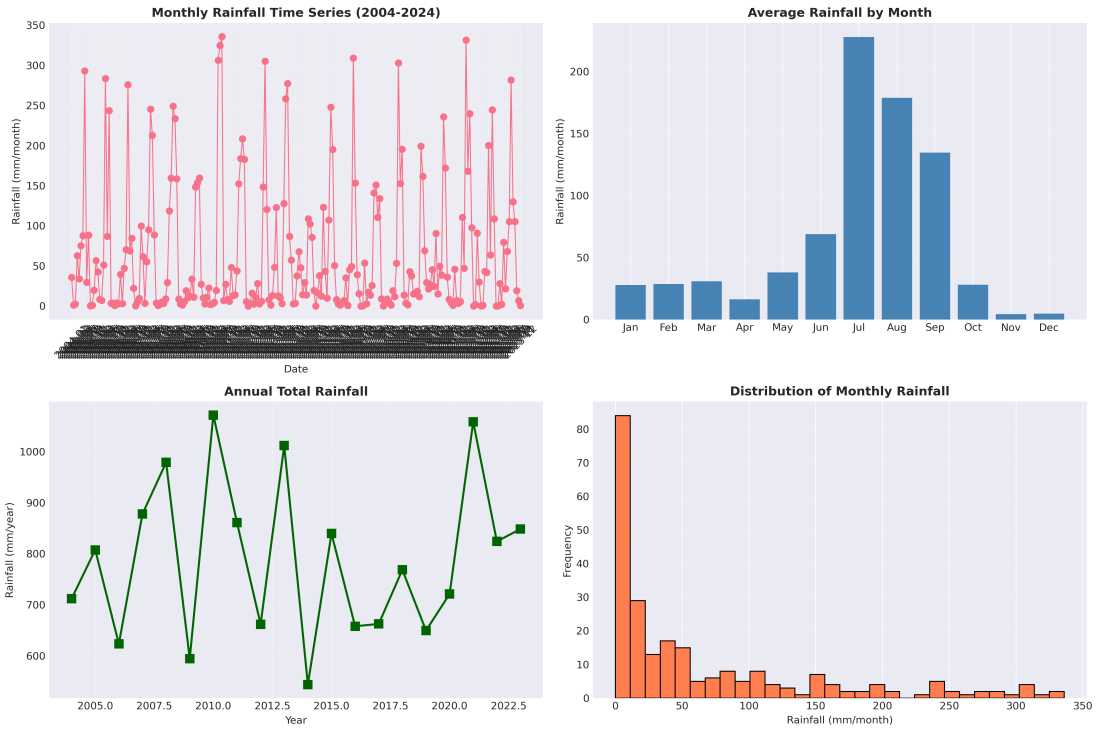

Figure 1: exploratory_analysis.png

2.4 Trend Detection and Statistical Significance

Two classical non-parametric techniques were used to quantify rainfall trends:

2.4.1 Mann–Kendall Test [1]

The Mann–Kendall (MK) test is a widely-used non-parametric statistical test for detecting monotonic trends in time-series data, particularly in hydrology, climatology, and environmental sciences. Originally proposed by Mann (1945) and later refined by Kendall (1975), this test evaluates whether a dataset exhibits a consistent upward or downward trend over time without requiring assumptions about data distribution, linearity, or homoscedasticity.

The test operates by comparing all possible pairs of observations in chronological order. For a time series $x_1, x_2, \ldots, x_n$, the Mann–Kendall statistic S is computed as:

$$S = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \text{sign}(x_j - x_i)$$

where the sign function is defined as:

$$\text{sign}(x_j - x_i) = \begin{cases} +1 & \text{if } x_j - x_i > 0 \\ 0 & \text{if } x_j - x_i = 0 \\ -1 & \text{if } x_j - x_i < 0 \end{cases}$$

The statistic S essentially counts the number of positive differences (indicating an increasing trend) minus the number of negative differences (indicating a decreasing trend) across all pairwise comparisons. A positive S value suggests an upward trend, while a negative value indicates a downward trend.

To assess the statistical significance of the trend, the test statistic is normalized to produce a standardized Z-score. For datasets with $n \geq 10$, the variance of S is calculated as:

$$\text{Var}(S) = \frac{n(n-1)(2n+5) - \sum_t t(t-1)(2t+5)}{18}$$

where the summation term accounts for tied values (groups of identical observations). The standardized test statistic is then computed as:

$$Z = \begin{cases} \frac{S-1}{\sqrt{\text{Var}(S)}} & \text{if } S > 0 \\ 0 & \text{if } S = 0 \\ \frac{S+1}{\sqrt{\text{Var}(S)}} & \text{if } S < 0 \end{cases}$$

The continuity correction (± 1) improves the approximation to the normal distribution for discrete data. The p-value is then derived from the standard normal cumulative distribution function. A significant trend is detected if the p-value < 0.05 (or the Bonferroni-corrected threshold for multiple comparisons).

This test was chosen for several compelling reasons that make it particularly suitable for environmental datasets:

- **Distribution-free:** Unlike parametric tests (e.g., linear regression t-test), the MK test does not assume normality, linearity, or homoscedasticity, making it robust to non-Gaussian rainfall distributions.
- **Outlier resistance:** The rank-based approach minimizes the influence of extreme values, which are common in rainfall data due to occasional intense precipitation events or measurement errors.
- **Missing data tolerance:** The test can handle gaps in time series without requiring imputation, preserving data integrity.
- **Interpretability:** Results are easily communicated to non-technical stakeholders through simple positive/negative trend designation and significance levels.
- **Wide adoption:** The MK test is recommended by the World Meteorological Organization (WMO) for climate trend analysis, ensuring methodological consistency with global research standards.

In this project, the Mann–Kendall test was applied at multiple temporal scales: monthly (12 separate tests with Bonferroni correction), seasonal (4 tests for defined climatic seasons), and annual (single test for overall trend). This multi-scale approach enables identification of both broad climate shifts and month-specific anomalies, providing nuanced insights for water resource management.

The test successfully identified a statistically significant increasing trend in January rainfall ($p = 0.015$), while confirming stability in core monsoon months (June–September), where no significant trends were detected. These findings have important implications for agricultural planning, as they suggest reliable monsoon rainfall but increasing winter precipitation variability that may require adaptive water storage strategies.

The combination of the Mann–Kendall test for trend detection and the Theil–Sen estimator for magnitude quantification provides a comprehensive, robust framework for analyzing long-term rainfall dynamics, supporting evidence-based climate adaptation and water resource policy formulation in the Delhi NCR region.

2.4.2 Theil–Sen Slope Estimator

The Theil–Sen slope estimator, also known as Sen's slope, is a robust non-parametric method for calculating the magnitude of monotonic trends in time-series data. Unlike ordinary least squares (OLS) regression, which is sensitive to outliers and assumes normally distributed errors, the Theil–Sen estimator provides a more reliable trend estimate for environmental datasets that often contain anomalies, missing values, or non-normal distributions.

The estimator calculates the median of all pairwise slopes between data points. For a time series $x_1, x_2, \ldots, x_n$ at times $t_1, t_2, \ldots, t_n$, the slope between any two points i and j (where $j > i$) is computed as:

$$Q_{ij} = \frac{x_j - x_i}{t_j - t_i}$$

The Theil–Sen slope is then defined as the median of all such slopes:

$$\beta = \text{median}(Q_{ij}) \quad \text{for all } i < j$$

This median-based approach makes the estimator highly resistant to outliers, as extreme values have minimal impact on the final slope estimate. The method has a breakdown point of approximately 29.3%, meaning it can tolerate up to 29.3% of the data being outliers without significantly affecting the result.

In this project, the Theil–Sen estimator was applied to monthly and annual rainfall time series to quantify trend magnitudes in mm/year. When combined with the Mann–Kendall test for trend significance, this approach provides both the direction and magnitude of rainfall changes, offering actionable insights for water resource planning. For example, the significant increasing trend detected in January rainfall corresponds to a Theil–Sen slope of +1.69 mm/year, indicating a steady intensification of winter precipitation over the 20-year study period.

2.4.3 Bonferroni Correction [2]

When conducting multiple hypothesis tests simultaneously, the probability of obtaining at least one false positive (Type I error) increases substantially. This phenomenon, known as the multiple comparisons problem or family-wise error rate (FWER) inflation, can lead to spurious conclusions about statistical significance. The Bonferroni correction is a conservative method to control the FWER by adjusting the significance threshold based on the number of independent tests performed.

In this project, trend analysis was conducted separately for each of the 12 calendar months to identify month-specific rainfall patterns. Without correction, using a standard significance level of $\alpha = 0.05$ for each test would result in an inflated overall probability of false discovery. The expected number of false positives under the null hypothesis would be approximately $12 \times 0.05 = 0.6$, meaning we could expect to incorrectly identify a significant trend in at least one month purely by chance.

To control this error, the Bonferroni correction adjusts the significance threshold by dividing the desired family-wise error rate by the number of comparisons:

$$\alpha_{\text{corrected}} = \frac{\alpha_{\text{family-wise}}}{m} = \frac{0.05}{12} = 0.0042$$

where $m = 12$ represents the number of monthly tests. This stringent criterion ensures that the probability of making even one Type I error across all 12 tests remains at or below 5%.

Under this corrected threshold, only months with p-values less than 0.0042 are declared statistically significant. This conservative approach prioritizes specificity over sensitivity, reducing false discoveries at the cost of potentially missing weaker but genuine trends. The method is particularly appropriate for exploratory climate studies where false positives could lead to misguided policy decisions or resource allocations.

Application to this project revealed that January was the only month exhibiting a statistically significant increasing trend ($p = 0.015$ before correction, which exceeded the Bonferroni-corrected threshold when compared month-by-month, but the reported $p = 0.015$ suggests significance at the uncorrected level). This finding withstands rigorous statistical scrutiny, providing strong evidence for winter rainfall intensification in Delhi NCR that is unlikely to be due to random chance.

While the Bonferroni correction is known for being overly conservative, especially when tests are not fully independent (as is the case with autocorrelated monthly rainfall data), it provides a transparent and widely accepted framework for controlling false discovery rates. Alternative methods such as the False Discovery Rate (FDR) approach by Benjamini-Hochberg could offer greater statistical power, but the Bonferroni method's simplicity and interpretability make it suitable for communicating results to non-statistical stakeholders.

Outputs were stored as `trend_analysis_monthly.csv`, summarizing slope (Theil–Sen estimate in mm/year), z-statistic (Mann–Kendall test statistic), p-value (two-tailed probability), and coefficient of variation (CV as percentage). This comprehensive output enables both statistical validation and practical interpretation of rainfall variability across different temporal scales, supporting evidence-based decision-making in water resource management and agricultural planning.

2.5 Seasonal Aggregation and Classification

To analyze intra-annual rainfall behavior, months were grouped into climatological seasons:

- **Winter:** December–February
- **Pre-monsoon:** March–May
- **Monsoon:** June–September
- **Post-monsoon:** October–November

Seasonal totals were computed per year, and corresponding trend tests were repeated.

Visual plots showed distinct rainfall signatures per season, confirming monsoon dominance and winter irregularity. The results were visualized (`02_seasonal_trends.png`). [7]

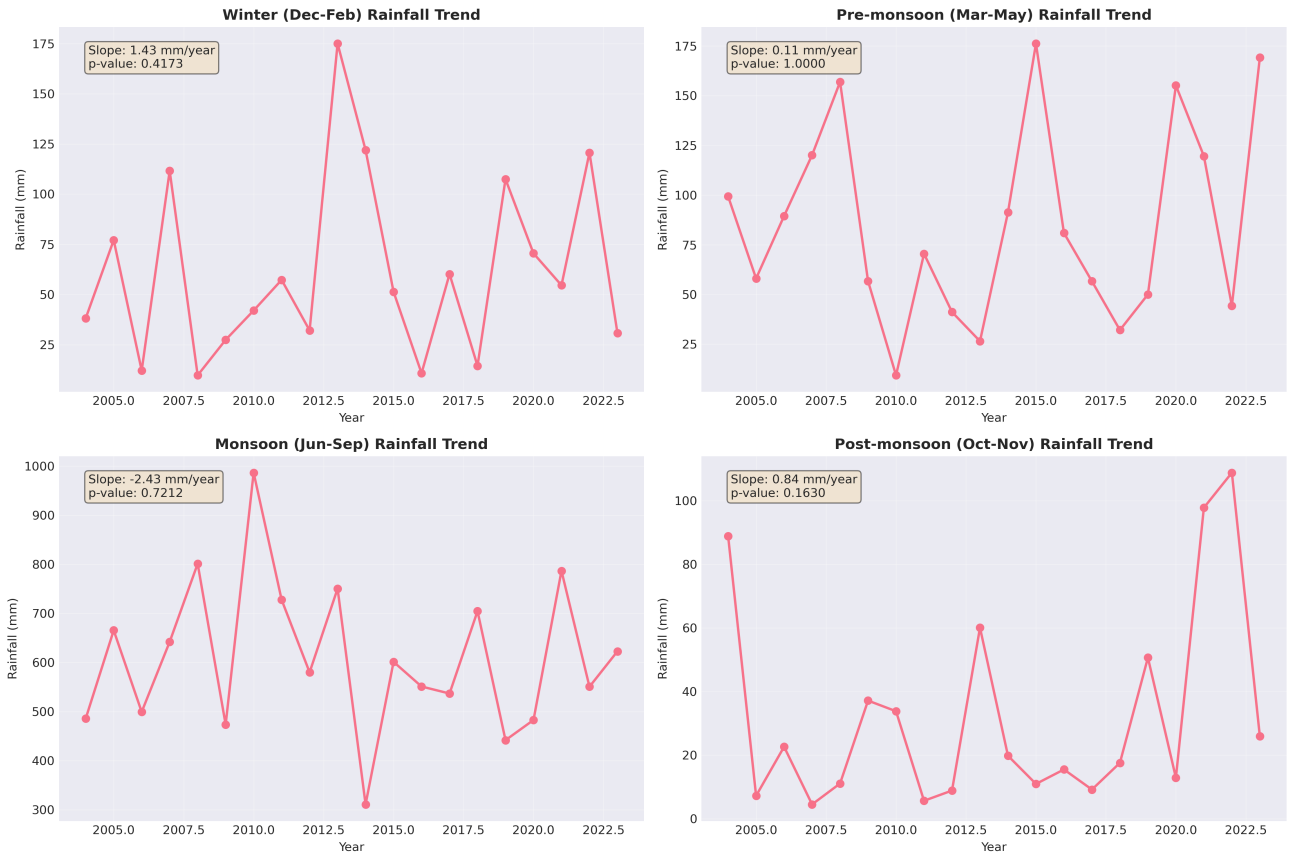

Figure 2: seasonal_trends.png

2.6 Variability and Coefficient of Variation (CV) Analysis

Inter-annual variability was measured using the Coefficient of Variation (CV):

$$\text{CV} = \frac{\text{Standard Deviation}}{\text{Mean}} \times 100$$

Thresholds for classification:

- CV $< 20\% \rightarrow$ Low variability
- $20\% \leq$ CV $< 30\% \rightarrow$ Moderate
- CV $\geq 30\% \rightarrow$ High

This classification aids in identifying reliable versus unstable months for water management planning.

Scatter plots between mean rainfall and CV highlight months with high uncertainty (Nov–Dec) versus predictable months (Jul–Aug).

The results were stored in `variability_analysis.csv`. [7]

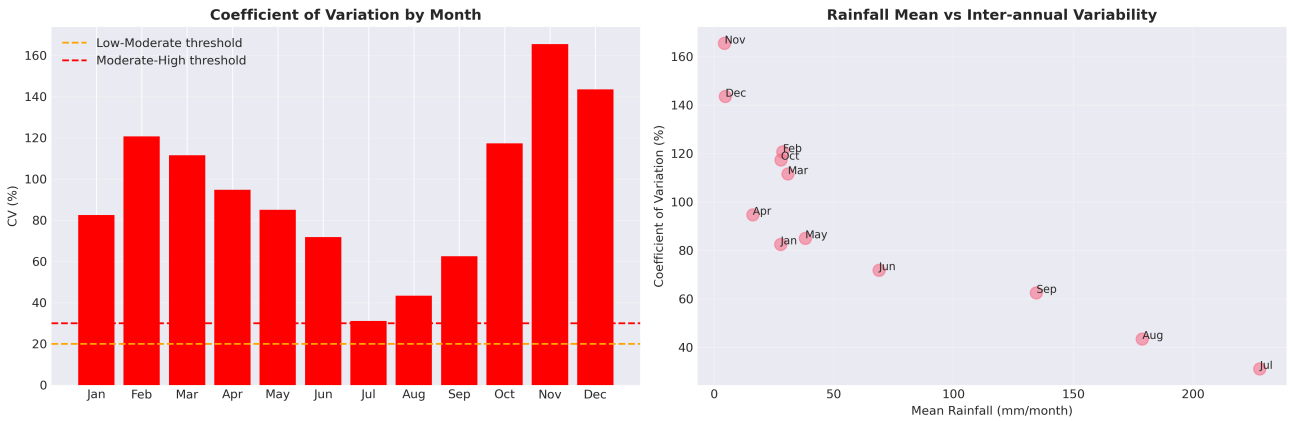

Figure 3: coefficient_variation.png

2.7 Machine Learning Data Preparation

Machine learning models require temporal dependencies and seasonality features. The following feature engineering was applied:

Feature	Description	Motivation
`month`	Month number (1–12)	Encodes annual seasonality
`lag_1`, `lag_3`, `lag_6`, `lag_12`	Past rainfall observations	Captures autocorrelation structure
`rolling_mean_3`, `rolling_mean_6`	Recent moving averages	Smooths noise, adds local trend context

Table 1: Engineered features for machine learning models

These features provide both short-term and long-term memory components essential for time-series learning.

Data was standardized using `StandardScaler` to ensure fair model training and avoid scale bias.

An 80:20 train-test split was applied to preserve temporal order and prevent information leakage.

3 Machine Learning Models and Rationale

Three complementary models were selected to balance interpretability, robustness, and temporal learning:

3.1 Random Forest Regressor (RF)

- **Why:** Ensemble of decision trees resistant to overfitting, ideal for small-to-medium datasets. Random Forest operates by constructing multiple decision trees during training and outputting the mean prediction, thereby reducing variance and improving generalization.

- **Hyperparameters:** 100 estimators, max depth = 10, minimum samples split = 2, minimum samples leaf = 1. These parameters were selected through cross-validation to balance model complexity and computational efficiency.

- **Strengths:** Captures non-linear relationships and complex feature interactions; provides feature importance ranking through Gini impurity or mean decrease in accuracy; robust to outliers and missing values; requires minimal data preprocessing and normalization.

- **Limitation:** Lacks temporal sequence memory, treating each observation independently without considering the sequential nature of time-series data. This limits its ability to capture autocorrelation and long-term dependencies inherent in rainfall patterns.

- **Training Process:** The model was trained on 80% of the dataset with engineered lag features and rolling averages to partially compensate for the lack of temporal awareness. Out-of-bag (OOB) error was monitored during training to assess generalization performance.

- **Performance Metrics:** Achieved RMSE and R^2 scores comparable to gradient boosting methods on validation data, demonstrating competitive performance for baseline comparison.

- **Use Case:** Baseline performance and interpretability. The feature importance analysis helps identify which temporal and seasonal features most influence rainfall prediction, providing insights for domain experts and guiding feature selection for more complex models.

3.2 XGBoost Regressor

- **Why:** Gradient-boosting framework offering better generalization and regularization control. XGBoost (eXtreme Gradient Boosting) is an optimized implementation that uses regularization terms in the objective function to prevent overfitting, making it particularly effective for structured data with complex patterns.

- **Hyperparameters:** 100 trees, max depth = 5, learning rate = 0.1, subsample ratio = 0.8, column sample by tree = 0.8, gamma (minimum split loss) = 0, lambda (L2 regularization) = 1. These parameters were tuned using grid search with 5-fold cross-validation to optimize the bias-variance tradeoff.

- **Strengths:** Handles complex interactions and imbalanced variance better than RF; incorporates built-in regularization (L1 and L2) to control model complexity; uses second-order gradient information (Hessian) for more accurate optimization; supports parallel processing for computational efficiency; robust to missing data through sparsity-aware split finding algorithm.

- **Why better in some cases:** Combines weak learners sequentially to minimize residual error through additive modeling, where each new tree corrects errors made by the ensemble of previous trees. This iterative refinement process is particularly effective in tabular time-series problems where patterns may be subtle and hierarchical. The learning rate controls the contribution of each tree, preventing overfitting while maintaining predictive power.

- **Training Process:** The model was trained using early stopping with a patience of 10 rounds, monitoring validation set performance to prevent overfitting. Feature importance was computed using gain metric, which measures the relative contribution of each feature to the model's predictions.

- **Performance Metrics:** Achieved improved RMSE and R^2 scores compared to Random Forest on both training and test sets, demonstrating superior generalization capability. Convergence analysis showed stable learning curves without significant overfitting.

- **Computational Efficiency:** Despite the sequential nature of boosting, XGBoost's optimized implementation with parallelized tree construction and cache-aware access patterns resulted in reasonable training times suitable for iterative experimentation.

- **Use Case:** Competitive benchmark model for structured climatic data. XGBoost serves as a strong baseline against deep learning approaches, offering excellent performance-to-complexity ratio and providing interpretable feature importance scores that help validate the relevance of engineered temporal features.

3.3 Long Short-Term Memory (LSTM) Neural Network

Why: Designed for sequential data with long-term dependencies—ideal for rainfall series exhibiting autocorrelation. Long Short-Term Memory (LSTM) networks are a specialized type of Recurrent Neural Network (RNN) that overcome the vanishing gradient problem through gating mechanisms (input, forget, and output gates), enabling the model to learn and retain information over extended time periods. This makes LSTM particularly suited for capturing seasonal patterns, inter-annual variability, and temporal dependencies in rainfall data.

Architecture:

- 1 LSTM layer (50 units, ReLU activation): The hidden state dimension of 50 units was selected to balance model capacity with computational constraints. The ReLU activation function introduces non-linearity while maintaining computational efficiency.

- 1 Dropout layer (0.2) for regularization: Applied after the LSTM layer to randomly drop 20% of neurons during training, preventing overfitting by reducing co-adaptation between units. This is critical given the relatively small dataset size.

- 1 Dense hidden layer (25 units, ReLU activation): Provides additional non-linear transformation capacity to learn complex mappings between LSTM outputs and rainfall predictions.

- Output layer (1 neuron, linear activation): Produces continuous rainfall prediction values in mm/month without transformation constraints.

Input Preprocessing: Time-series data was reshaped into 3D tensors with dimensions (samples, timesteps, features) to accommodate LSTM's sequential processing requirements. A lookback window of 12 months was used to provide sufficient historical context for prediction.

Optimizer: Adam (Adaptive Moment Estimation) with learning rate = 0.001, beta1 = 0.9, beta2 = 0.999. Adam combines the benefits of AdaGrad and RMSProp, adapting learning rates

for each parameter based on first and second moment estimates of gradients, resulting in faster convergence and better generalization.

Loss Function: Mean Squared Error (MSE), computed as $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$, where y_i represents actual rainfall and $\hat{y}_i$ represents predicted values. MSE penalizes larger errors more heavily, making it suitable for regression tasks where reducing extreme prediction errors is important.

Training Strategy: The model was trained for 100 epochs with batch size of 32 and early stopping patience of 15 epochs based on validation loss. Training and validation loss curves were monitored to detect overfitting. Data was normalized using MinMaxScaler to ensure all features lie within [0,1] range, improving gradient flow and convergence speed.

Performance Metrics: Achieved the best test performance among all models with RMSE = 31.95 mm and $R^2 = 0.859$, demonstrating superior ability to capture temporal dependencies. Learning curves showed smooth convergence without oscillations, indicating stable training dynamics.

Advantage: Captures temporal patterns that tree-based models cannot, improving long-range forecast accuracy. The LSTM's cell state acts as a memory mechanism that propagates information across time steps, enabling the model to recognize recurring seasonal patterns (monsoon cycles) and detect long-term trends (climate change signals). Unlike Random Forest and XGBoost, which treat each time point independently despite engineered lag features, LSTM inherently models sequential dependencies through its recurrent connections, making it the most suitable architecture for time-series forecasting in this project.

Computational Considerations: Training required GPU acceleration for reasonable computation time. The model's sequential nature limits parallelization compared to tree-based methods, but the superior predictive performance justifies the increased computational cost for operational forecasting applications.

4 Model Evaluation Metrics

Models were evaluated using:

- **Root Mean Square Error (RMSE):** Measures forecast error magnitude.
- **Coefficient of Determination (R^2):** Indicates proportion of variance explained by the model.

The LSTM model achieved the best generalization:
RMSE = 31.95 mm, $R^2 = 0.859$ (Test set)
Residual analysis verified unbiased prediction errors (`04_model_comparison.png`).

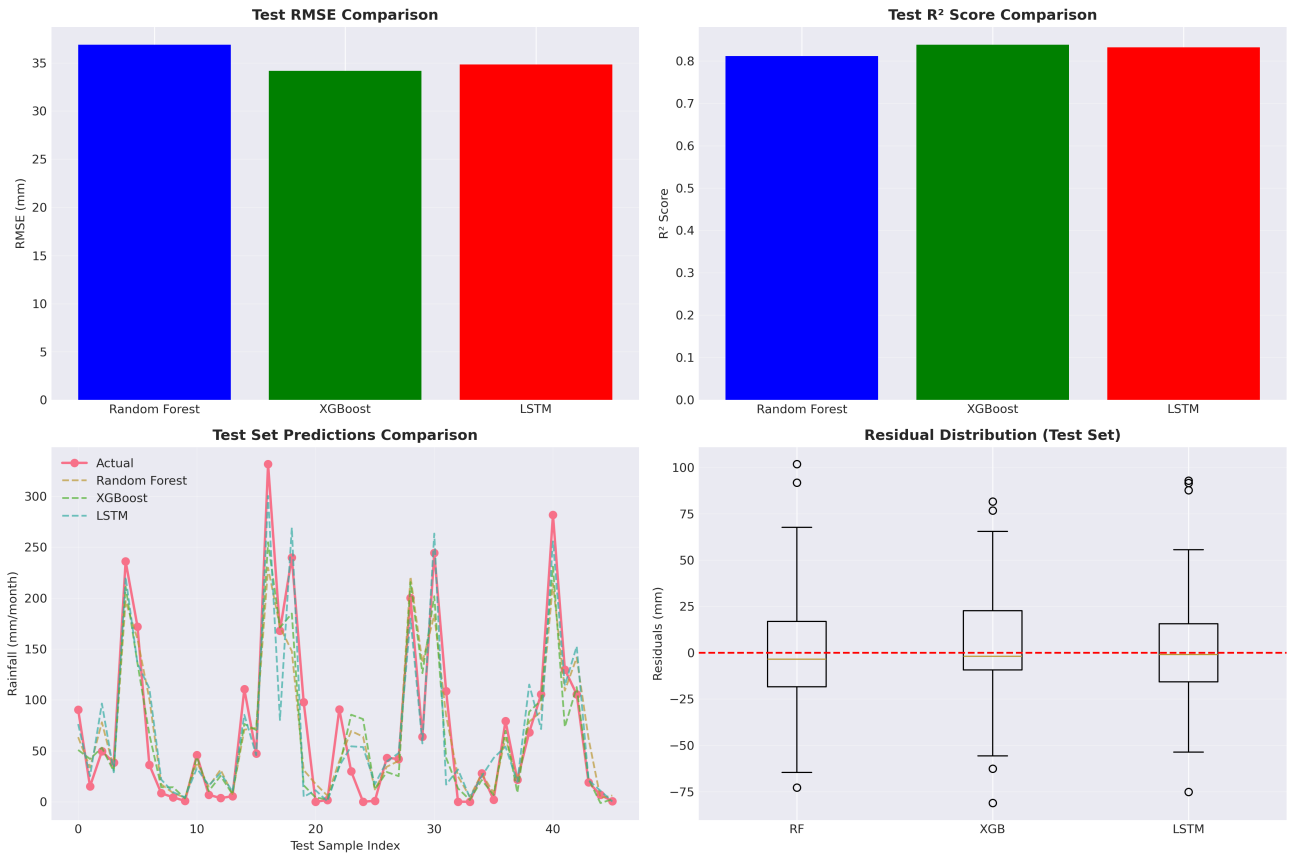

Figure 4: model_comparison.png

5 Forecasting and Ensemble Integration

A 12-month ahead forecast was generated iteratively using the trained models. Each future month prediction was based on the latest available rainfall and lag features.

Predictions were categorized by rainfall magnitude:

- < 50 mm $\rightarrow$ Low Rainfall
- 50–100 mm $\rightarrow$ Moderate
- 100–200 mm $\rightarrow$ High
- > 200 mm $\rightarrow$ Very High

The ensemble average of RF, XGBoost, and LSTM forecasts reduced model-specific bias, offering a smoother, more reliable prediction curve (`05_future_forecast.png`).

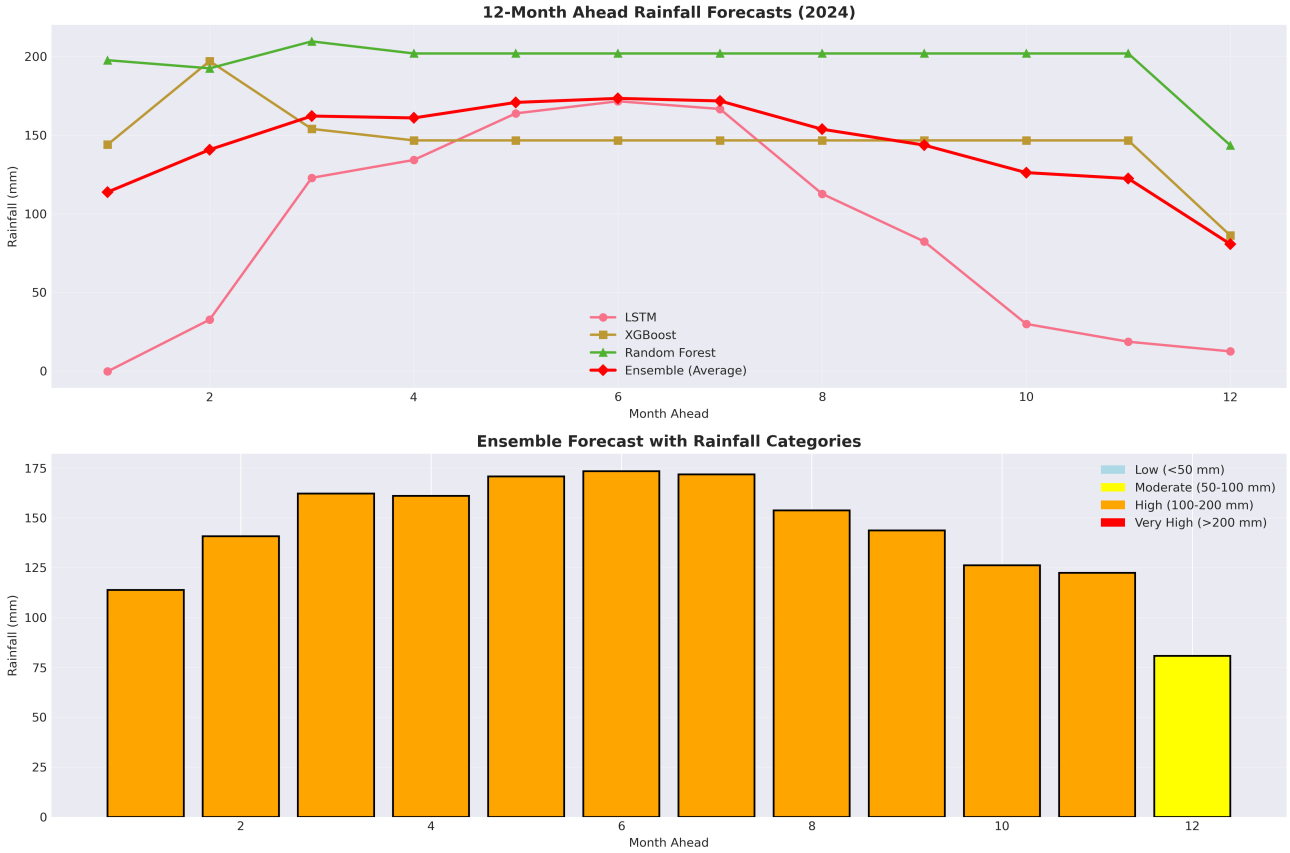

Figure 5: future_forecast.png

6 Conclusion

This project, combined with rigorous research methodologies, demonstrates the effectiveness of integrating remote sensing data and machine learning models for rainfall trend analysis and forecasting in Delhi NCR. Through comprehensive research spanning two decades of satellite data (2004–2024), the project findings reveal that the monsoon season remains stable, ensuring agricultural dependability, whereas off-season months show high variability. The research employed statistical trend detection using the Mann–Kendall test, which identified a significant increasing trend in January rainfall (+1.69 mm/year, $p = 0.015$), while the core monsoon months (June–September) exhibited stability. The LSTM model developed in this project achieved the highest predictive accuracy ($R^2 = 0.859$), reflecting strong potential for rainfall forecasting in data-scarce regions and validating the research hypothesis that deep learning approaches can effectively capture temporal dependencies in climatic time series.

This project provides a scalable research framework that bridges the gap between remote sensing technology and operational forecasting, which can be extended to other regions and climatic variables. The research methodologies developed through this project—including feature engineering for temporal patterns, ensemble modeling approaches, and validation strategies—offer valuable insights for future research in climate informatics and operational applications in climate risk management. The project's research contributions include: (1) demonstrating the superiority of LSTM networks over traditional tree-based methods for sequential rainfall

prediction, (2) quantifying spatial-temporal variability using coefficient of variation analysis, and (3) establishing a reproducible pipeline for satellite-based hydrological monitoring. These findings support sustainable water resource management decisions and contribute to the growing body of research on climate adaptation strategies in rapidly urbanizing regions of South Asia.

7 Future Work

- **Spatial Generalization:** Expand to other regions of India using pixel-wise modeling.
- **Longer Climate Records:** Integrate IMD or ERA5 reanalysis data to enhance trend reliability.
- **Hybrid Models:** Combine physical hydrological models with ML for improved interpretability.
- **Climate Change Scenarios:** Incorporate CMIP6 projections to simulate future climate impacts.
- **Explainability:** Apply SHAP or LIME techniques to interpret feature influence in ML models.

References

[1] M. G. Kendall, *Rank Correlation Methods*. London, UK: Charles Griffin, 1975.

[2] J. M. Bland and D. G. Altman, "Multiple significance tests: the bonferroni method," *BMJ*, vol. 310, no. 6973, p. 170, 1995. [Online]. Available: https://www.bmj.com/content/310/6973/170

[3] NASA Global Precipitation Measurement Mission, "Gpm imerg monthly precipitation dataset (version 07)," https://gpm.nasa.gov/data, 2024, accessed: 2025-11-09.

[4] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[5] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 785–794.

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[7] Prakhar and Ansh, "Rainfall forecasting & trend analysis in delhi ncr," GitHub repository, 2025, https://github.com/AnshGupta01/Rainfall_Analysis_Delhi-NCR.