**Methodology for Attention-Based LSTM Rainfall Prediction Model Using GCM and IMD Data**

### **1. Data Loading and Preprocessing**

#### **a. Observed IMD Data**

* Dataset: rainfall\_imd\_monthly.csv
* Monthly rainfall data per grid (4641 grids across India).
* Date column parsed into numerical **Month** and **Year**.

#### **b. Train-Test Split**

* **Training Set**: 2015 to 2021.
* **Testing Set**: 2022 to 2023.

#### **c. Enhanced Seasonal Features**

* **sin(month \* 2π / 12)**: captures seasonality.
* **cos(month \* 2π / 12)**: phase complement.
* **Monsoon Flag**: binary feature (1 if June–September).
* **Quarter Encoding**: Q1 to Q4 indicators.

### **2. Normalization**

* **Technique**: RobustScaler (resistant to outliers).
* Applied to observed IMD data.
* Reshaped format: (n\_months, 4641) → each month as a vector of grid rainfall values.

### **3. GCM Ensemble Processing**

* Multiple GCM predictions loaded from .csv files.
* For each GCM:  
  + Aligned with IMD years.
  + Shapes matched (truncated if needed).
  + Normalized using the same scaler as IMD.
* **Ensemble Mean**:  
  + mean\_gcm\_train: average of all GCMs for training years.
  + mean\_gcm\_test: for test years.

### **4. Enhanced TIF (Truth-Indeterminacy-Falsehood) Calculation**

* Computed per grid point (4641 total):  
    
   **Truth**:  
  + Accuracy-focused: MSE + Correlation + NSE
* **Indeterminacy**:  
  + Error spread: MAE + Max deviation
* **Falsehood**:  
  + Pattern mismatch: MSE + temporal inconsistency
* Visual Outputs:  
  + Histograms, scatter plots, and heatmaps of reliable vs noisy grid points.

### **5. TIF-Based Weighting Strategies**

| **Weighting Name** | **Formula** |
| --- | --- |
| Simple | Truth / sum(Truth) |
| Balanced | (Truth + (1 - Falsehood)) / sum(...) |
| Advanced | 2 \* Truth \* (1 - Falsehood) |
| Nonlinear | Truth^3 / sum(Truth^3) |

* These weights are applied to the GCM ensemble per grid.
* Best-performing scheme: **Nonlinear** (correlation ~0.5213).

### **6. Model Input Preparation**

* Three time-series input sequences prepared per sample:  
    
   **(a) Observed Sequence (X\_observed\_train)**
  + Shape: (samples, sequence\_length, 4641)
  + Normalized rainfall per grid
* **(b) GCM Sequence (X\_train)**
  + Shape: (samples, sequence\_length, 4641)
  + TIF-weighted ensemble GCM data
* **(c) Seasonal Features (seasonal\_observed\_train)**
  + Shape: (samples, sequence\_length, 4)
  + sin, cos, monsoon, quarter indicators
* Combined Input Shape: (samples, sequence\_length, ~9286)

### **7. Model Architecture**

* **Input Layer**: (12, ~9286) → 12-month sequences
* **Bidirectional LSTM**: captures temporal dynamics
* **Attention Layer**: assigns importance to time steps
* **Dense Output Layer**: predicts rainfall for 4641 grids

### **8. Model Training**

* model.fit(...) with:  
  + 20% validation split
  + Callbacks: Early stopping + LR reduction

### **9. Forecasting Logic**

* Autoregressive multi-step:  
  + Use past 12 months to predict next month
  + Slide window forward, repeat recursively
* Smart fallback: use predictions when GCM is missing

### **10. Inverse Scaling**

* Rescale predicted values back to rainfall units (mm).
* Enables interpretation and real-world metric evaluation.

### **11. Evaluation Metrics**

**Standard Metrics**:

* MSE, RMSE, MAE, R^2

**Hydrological Metrics**:

* NRMSE: Normalized RMSE
* MAPE: Mean Absolute Percentage Error
* NSE: Nash-Sutcliffe Efficiency
* KGE: Kling-Gupta Efficiency

### **12. Visual Diagnostics**

* Time-series comparison plots (observed vs predicted)
* Scatter plots with correlation overlays
* Monthly average comparisons
* Seasonal breakdown (monsoon vs non-monsoon)

### **13. Effectiveness of TIF Approach**

* TIF improves spatial filtering:  
  + Promotes reliable (high T) regions
  + Suppresses noisy (high F) areas
* Leads to better ensemble quality and LSTM performance

This end-to-end methodology ensures robust rainfall forecasting by integrating climate model ensembles with uncertainty quantification (TIF) and sequence learning via attention-equipped L

Here is the **complete methodology with detailed step-by-step inputs and outputs** after each phase of your rainfall prediction pipeline, explaining data shapes, flow, and purpose.

## **🌧️ Rainfall Prediction Model: Full Methodology with Inputs & Outputs**

### **1. Data Loading and Preprocessing**

**Input:**

* rainfall\_imd\_monthly.csv: Monthly observed rainfall over India, shape = (n\_months, 4641)  
   → Each column is a spatial grid (1x1 degree), each row a monthly timestep.

**Process:**

* Extract Month and Year from column Date (e.g., "Jan-22" → Month = 1, Year = 2022)
* Filter and split into:  
  + Train: 2015–2021
  + Test: 2022–2023
* Generate seasonal features:  
  + sin\_month, cos\_month: Cyclical month features → shape = (n\_samples, 1)
  + monsoon\_flag: 1 if June–Sept, else 0 → (n\_samples, 1)
  + quarter: 1 to 4 → (n\_samples, 1)

**Output:**

* observed\_train: (84, 4641)
* observed\_test: (24, 4641)
* seasonal\_features\_train: (84, 4)
* seasonal\_features\_test: (24, 4)

### **2. Normalization**

**Input:**

* observed\_train, observed\_test

**Process:**

* Apply RobustScaler() across each spatial grid point (axis=0):  
  + Handles outliers better than MinMaxScaler or StandardScaler.
* Fit on training only; transform both sets.
* Optional reshaping for LSTM if required: (n\_samples, 4641) → (n\_samples, 4641, 1)

**Output:**

* normalized\_observed\_train: (84, 4641)
* normalized\_observed\_test: (24, 4641)

### **3. GCM Ensemble Processing**

**Input:**

* Multiple GCM .csv files, each: (n\_months, 4641)

**Process:**

* Load each GCM file
* Align with IMD data years
* Apply same RobustScaler from IMD
* Stack all GCMs and compute ensemble mean:  
  + mean\_gcm\_train: (84, 4641)
  + mean\_gcm\_test: (24, 4641)

**Output:**

* Normalized and aligned:  
  + gcm\_train\_raw\_list: List[(84, 4641)]
  + gcm\_test\_raw\_list: List[(24, 4641)]
* Mean:  
  + mean\_gcm\_train: (84, 4641)
  + mean\_gcm\_test: (24, 4641)

### **4. Enhanced TIF Calculation**

**Input:**

* observed\_train (truth)
* gcm\_train\_raw\_list (predictions from models)

**Process:**

* For each grid (4641 total):  
  + Compute:  
    - Truth: weighted by NSE, Correlation, inverse MSE
    - Indeterminacy: high if MAE & max deviation large
    - Falsehood: high if pattern mismatches

**Output:**

* truth\_scores: (4641,)
* falsehood\_scores: (4641,)
* indeterminacy\_scores: (4641,)
* Visualizations: 2D plots, heatmaps, histograms

### **5. TIF Weighting Strategies**

**Input:**

* truth\_scores, falsehood\_scores

**Process:**

Apply 4 strategies:

| **Weight Name** | **Formula** |
| --- | --- |
| Simple | Truth / sum(Truth) |
| Balanced | (Truth + (1 - Falsehood)) / sum(...) |
| Advanced | 2 \* Truth \* (1 - Falsehood) |
| Nonlinear | Truth³ / sum(Truth³) |

* Apply weights to GCMs:  
  + mean\_gcm\_train\_weighted = mean\_gcm\_train \* weights
  + mean\_gcm\_test\_weighted = mean\_gcm\_test \* weights

**Output (for Nonlinear TIF weights):**

* weighted\_gcm\_train: (84, 4641)
* weighted\_gcm\_test: (24, 4641)

### **6. Sequence Preparation for LSTM**

**Input:**

* weighted\_gcm\_train, normalized\_observed\_train, seasonal\_features\_train

**Process:**

* Use a rolling window (12 months) to form input-output sequences:  
  + X\_observed: previous 12 months of observed data
  + X\_gcm: previous 12 months of GCM data
  + X\_seasonal: previous 12 months of seasonal features
  + Y: 1-month-ahead observed rainfall

**Output:**

* X\_observed\_train: (n\_samples, 12, 4641)
* X\_gcm\_train: (n\_samples, 12, 4641)
* X\_seasonal\_train: (n\_samples, 12, 4)
* Y\_train: (n\_samples, 4641)

### **7. LSTM Model Training**

**Input to Model:**

* X\_observed\_train: (n, 12, 4641)
* X\_gcm\_train: (n, 12, 4641)
* X\_seasonal\_train: (n, 12, 4)

**Model:**

* 3 Inputs → Attention + LSTM layers → Merge → Dense
* Uses callbacks (EarlyStopping, ReduceLROnPlateau)

**Output:**

* Trained model weights
* history: Training/validation loss logs

### **8. Autoregressive Forecasting (Test Phase)**

**Input:**

* Last 12 months of:  
  + observed\_test
  + weighted\_gcm\_test
  + seasonal\_test

**Process:**

* Predict one month at a time
* Append prediction to the rolling window
* Repeat for next month

**Output:**

* predicted\_test: (24, 4641)
* Predicted rainfall (normalized)

### **9. Inverse Scaling**

**Input:**

* predicted\_test, normalized\_observed\_test

**Process:**

* Apply inverse of RobustScaler to convert back to mm

**Output:**

* final\_predictions\_mm: (24, 4641)
* Real rainfall units

### **10. Metrics Calculation**

**Input:**

* final\_predictions\_mm
* observed\_test

**Process:**

* Compute:  
  + MSE, RMSE, MAE, R²
  + NRMSE, NSE, KGE, MAPE

**Output:**

* Metrics dictionary with values

### **11. Visual Diagnostics**

**Input:**

* final\_predictions\_mm, observed\_test

**Process:**

* Generate:  
  + Line plots (time series)
  + Scatter plots (pred vs obs)
  + Seasonal averages
  + Correlation maps

**Output:**

* plots/: saved figures
* Inline metrics and plots for analysis

### **12. Seasonal Performance Analysis**

**Input:**

* final\_predictions\_mm, observed\_test, months

**Process:**

* Split into:  
  + Monsoon months (June–Sept)
  + Non-monsoon months
* Compute metrics separately

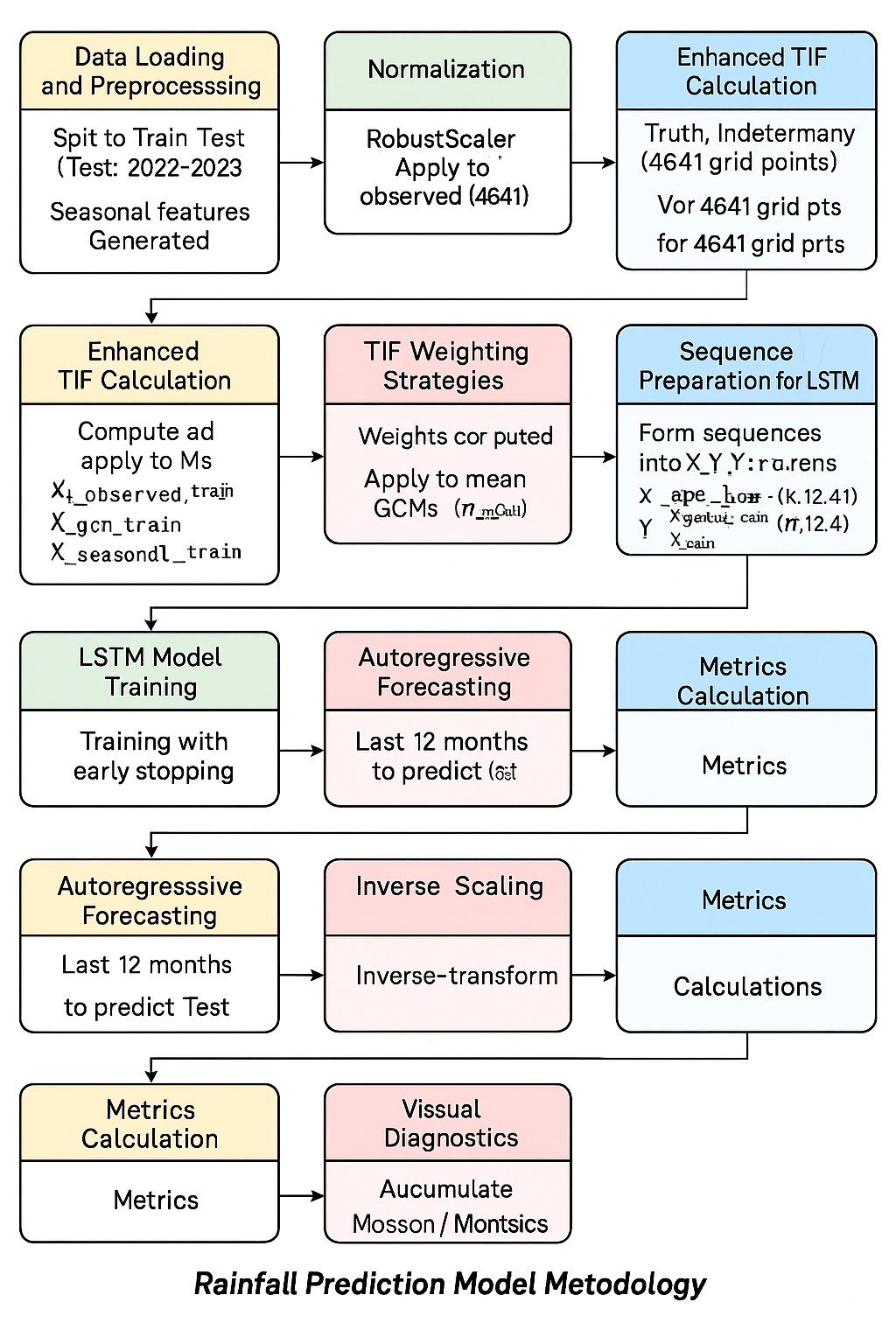
**Output:**

* seasonal\_metrics: Performance for each seasonal bucket

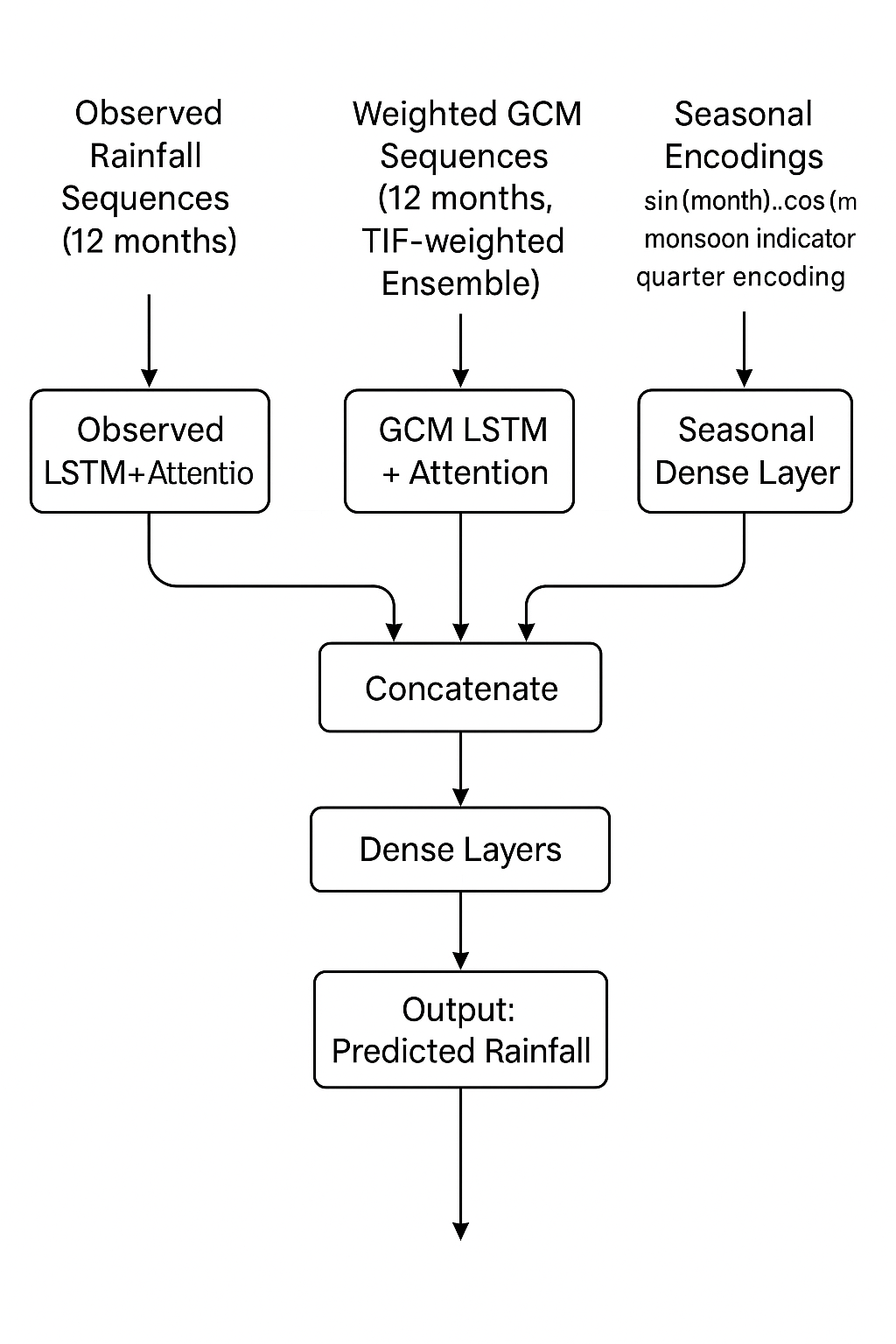
### **✅ Summary of Input/Output Flow to LSTM:**

| **Component** | **Shape** | **Notes** |
| --- | --- | --- |
| X\_observed\_train | (n, 12, 4641) | Past observed rainfall |
| X\_gcm\_train | (n, 12, 4641) | Past TIF-weighted GCM |
| X\_seasonal\_train | (n, 12, 4) | Seasonal encodings |
| Y\_train | (n, 4641) | Target rainfall after 12 months |

* Diagram of this flow



* Model architecture visual



* Annotated code walkthrough matching this step-by-step methodology

MODEL ARCHITECTURE

