SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Monthly Rainfall Prediction of Kerala Using Lag-Llama And Lstm

Asha J[1], Santhoshkumar S[2], Rishidas S[3], Viji R.[4], Resmi R[5]

[1]Department of Electronics & Communication Engg., Affliated to A.P.J Abdul Kalam Kerala Technological University, Government Engineering College, Thrissur-680009, Kerala, India. E-mail: ashaj@gectcr.ac.in
[2]Department of Electronics & Communication Engg., Affliated to A.P.J Abdul Kalam Kerala Technological University, Government Engineering College, Idukki-685603, Kerala, India. E-mail: santhoshs@gecidukki.ac.in,
[3]Department of Electronics & Communication Engg. ,Affliated to A.P.J Abdul Kalam Kerala Technological University, Government Engineering College, Barton Hill-695035, Kerala, India. E-mail: rishidas19731999@gmail.com
[4]Department of Electronics & Communication Engg., Affliated to A.P.J Abdul Kalam Kerala Technological University, College of Engineering Trivandrum, Thiruvananthapuram-695016, Kerala, India. E-mail: vijir@cet.ac.in
[5]Department of Electronics & Communication Engg. ,Affliated to A.P.J Abdul Kalam Kerala Technological University, College of Engineering Trivandrum, Thiruvananthapuram-695016, Kerala, India. E-mail: resmi@cet.ac.in

ABSTRACT
Accurate monthly rainfall prediction is important for climate resilience and water resource management in Kerala, India. This study presents a novel rainfall forecasting methodology using the Lagged Language Meta AI (Lag-LLama) model, an open-source foundation model designed specifically for time series forecasting. The proposed approach applies Lag-LLama for monthly rainfall prediction and evaluates its performance against established machine learning and deep learning models, namely Random Forest (RF) and Long Short-Term Memory (LSTM) networks.
Results show that while RF and LSTM outperform traditional statistical methods, the fine-tuned Lag-LLama model achieves a substantial improvement of approximately 41–42% in prediction accuracy over its zero-shot configuration and consistently surpasses both RF and LSTM models. These findings demonstrate the transformative potential of foundation models for complex climate time series forecasting and highlight the importance of domain-specific fine-tuning. The proposed methodology establishes a new benchmark for rainfall prediction in Kerala and offers a transferable framework for similar climatic regions under increasing climate variability.

Keywords: Lag Llama; Lstm; Monthly Rainfall; Kerala; Random Forest

INTRODUCTION

Kerala flanked by the Arabian Sea and the Western Ghats, receives heavy annual rainfall, between June and September. This rainfall is vital for agriculture, water resources, electrical energy production and ecological balance. The Southwest monsoon, falling between June and September, brings the heaviest rainfall, contributing around 70-80% of annual rainfall. Even though this rain is vital for agriculture and water resources, it also poses flood risks. The northeast monsoon between October and November contributes 10 to 15 % of the rainfall, especially to the southern parts of the state. Pre-monsoon showers between March and May provide around 10% of the rainfall, crucial for early farming activities. Winter is usually dry from December to February, with very little rain. Local topographical effects further complicate Kerala's monsoon patterns. The Western Ghats create strong orographic effects, leading to enhanced precipitation on windward slopes and rain shadow effects on leeward areas. The timing and amount of rainfall are crucial for cultivating major crops like rice, coconut, rubber, pepper, and banana. Monthly predictions enable decisions about irrigation, pesticide application, and crop protection. Reliable rainfall forecasts reduce crop losses due to erratic rainfall, thus ensuring food security and agricultural incomes.

This article discusses the monthly prediction of Kerala using the Lagged Language Meta AI (Lag-Llama) model. The proposed methodology for monthly rainfall prediction uses Lag-Llama. Lag-Llama is a significant advancement in time series forecasting, particularly applicable to complex time series predictions. As the first open-source foundation model specifically designed for time series forecasting, it offers powerful capabilities for both zero-shot predictions and fine-tuned applications in time series forecasting. The proposed work uses Lag-Llama's architecture for monthly rainfall prediction. It examines its architectural components and advantages over traditional forecasting methods by comparing the forecasting performance with Random Forest and Long Short-Term Memory (LSTM) prediction models.

Chattopadhyay and Chattopadhyay [1] compared ARIMA against ARNN Indian Summer Monsoon Rainfall

(ISMR) prediction. The seasonal ARIMA model was found to provide consistent and satisfactory predictions for monthly-scale rainfall parameters, with performance quantified using R^2 and RMSE metrics. Kamath and Kamath [2] compared three forecasting techniques—ARIMA, Artificial Neural Networks (ANN), and Exponential Smoothing (ETS)—using historical rainfall data from Idukki, Kerala to evaluate their predictive performance. Results indicated that the ARIMA model performed effectively when the rainfall series exhibited stable linear trends and seasonal structures, making it suitable for conventional time-series forecasting. In contrast, the ANN model showed superior capability in capturing complex nonlinear relationships and irregular rainfall variations, while ETS provided moderate performance for short-term trend estimation.

Wani et al. [3] address the challenge of rainfall prediction in the data-scarce North-Western Himalayas by benchmarking a broad suite of machine learning algorithms (RF, SVR, ANN, KNN), deep learning architectures (LSTM, Bi-directional LSTM, Deep LSTM, GRU, RNN), and classical time series techniques (ARIMA, TBATS) using meteorological records from six stations at varying elevations spanning 1980–2021. Their results show that deep learning methods achieve the highest accuracy as measured by RMSE and MAE — with Bi-directional LSTM ranking first — followed by ML algorithms, with ANN leading that group, and classical time series models performing least accurately. Notably, the study finds that altitude significantly influences model accuracy, underscoring the need for denser observational networks in mountainous terrain to improve forecast reliability.

Kumar, Kedam, Kisi et al. [4] conducted a comparative study of multiple machine learning models for daily and weekly rainfall forecasting to identify the most effective approach for short-term precipitation prediction. Multiple machine learning algorithms were evaluated, with CatBoost, XGBoost, and Random Forest performing best for daily forecasting, while XGBoost achieved the highest weekly prediction accuracy with an R^2 value of 0.99.

Akshaya et al. [5] presented an LSTM-based rainfall forecasting model for Kerala, focusing on monthly rainfall prediction in Thrissur, Pathanamthitta, Munnar, and Kottayam. The study demonstrated that LSTM networks effectively capture temporal dependencies and nonlinear rainfall patterns, making them suitable for complex climatic datasets. Model performance was assessed using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), confirming the effectiveness of deep learning for Kerala rainfall prediction. Saikia et al. [6] proposed a hybrid Wavelet-SARIMA-Transformer model for rainfall forecasting by integrating wavelet decomposition, SARIMA, and Transformer networks. In this framework, wavelet decomposition was used to separate rainfall series into multiscale components, SARIMA modeled linear and seasonal patterns, while the Transformer captured complex nonlinear dependencies. The study reported improved forecasting accuracy over standalone statistical and deep learning models, highlighting the effectiveness of hybrid approaches for rainfall prediction.

J. Adhikary and R. Maiti [7], developed long-term monthly rainfall forecasts for 19 districts of West Bengal using historical data from 1900–2019 and predicting the subsequent 108 months. The authors proposed a hierarchical spatio-temporal modeling framework that combined regression-based yearly feature forecasting with Multi-Layer Perceptron (MLP) networks to capture nonlinear temporal patterns and cross-district dependencies. Results showed that the proposed HSTM model consistently outperformed Naïve Seasonal and STLM models in terms of sMAPE and NRMSE, demonstrating the importance of integrating spatial interactions for regional rainfall forecasting.

Novitasari et al. [8] implemented a Backpropagation Neural Network (BPNN) utilizing infrared satellite data to enhance regional rainfall prediction accuracy. By tracking structural cloud markers and brightness temperatures, the model identified heavy rain-bearing clouds over Indonesia with 88.28% accuracy. The study highlights the effectiveness of using high-resolution, satellite-derived data for improving meteorological forecasting models.

Although previous studies have successfully applied ARIMA, Random Forest, ANN, and LSTM models for rainfall prediction in India and Kerala, the application of foundation time-series models such as Lag-LLaMA for monthly rainfall forecasting in Kerala remains largely unexplored. Existing approaches have limitations in modeling long-term dependencies, statewide climatic variability, and forecast uncertainty. Therefore, this research work aims to evaluate Lag-LLaMA as an advanced forecasting framework for Kerala monthly rainfall prediction and compare its performance with traditional ML and deep learning models.

The remaining part of the paper is organized as follows. Section 2 investigates the methodologies employed for monthly rainfall prediction. Section 3 elaborates on the dataset used. Section 4 includes results and discussions.

2.Methodology

2.1. Lag-Llama

Deep learning approaches to forecasting are emerging as powerful alternatives to traditional statistical benchmark models like ARIMA. The field is witnessing the development of foundation models for time series data that operate on principles similar to those of other generative Artificial Intelligence systems. The ML field is experiencing a revolution driven by the emergence of foundation models. The expansive, versatile neural networks undergo pretraining through unsupervised learning techniques on diverse, large-scale datasets spanning multiple distributions.

These models exhibit exceptional ability to generalise with minimal examples. They frequently perform better than models specifically designed for individual datasets when applied to various forecasting tasks. This few-shot learning capability significantly advances how ML systems can rapidly adapt to new problems with limited training examples.

These specialised models undergo training on extensive time series datasets, enabling them to generate deterministic and probabilistic forecasts. Their ability to produce estimates without task-specific pretraining makes time series foundation models particularly valuable. They operate similarly to large language models (LLMs) that can create text without being explicitly trained for specific text generation tasks. Large Language Models (LLMs) represent a significant advance in AI because they utilise neural networks with many parameters to interpret language more effectively. This capability significantly advances AI-powered forecasting technology, offering new paradigms for time series analysis across various domains.

Lag-LLama adapts the transformer-based architecture popularised by large language models to the unique challenges of time series data [9]. Unlike traditional forecasting models that require extensive feature engineering, Lag-LLama leverages self-attention mechanisms to capture complex temporal dependencies across multiple horizons and variables.

The model employs a "lag-based" approach, where historical time points are encoded to allow the model to learn relevant patterns across different temporal scales. This architecture enables Lag-LLama to process multivariate time series data effectively, handling long-term trends and seasonal patterns [10].

Lag-LLama integrates transformer-based deep learning with specialised components for time series forecasting, creating a robust foundation model capable of probabilistic predictions across diverse domains. Fig. 1 shows the general block diagram of Lag-LLama that is used for time series forecasting.

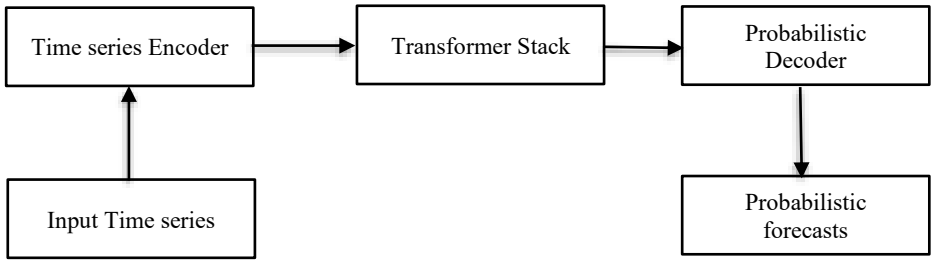

Figure 1: Proposed prediction model using Lag-Llama

The lag-based encoder transforms raw time series into a format highlighting temporal dependencies at various scales, making them accessible to the transformer architecture.

$$\text{Original Time Series:} \quad [X_1, X_2, X_3, \ldots\ldots, X_t]$$

$$\text{Lag encoding (forecast point } t\text{)}$$

$$\text{Recent Lags:} \quad [X_{t-1}, X_{t-2}, X_{t-3}]$$

$$\text{Seasonal Lags:} \quad [X_{t-7}, X_{t-14}, X_{t-21}]$$

$$\text{Long-term lags:} \quad [X_{t-30}, X_{t-60}, X_{t-90}]$$

Lag-Llama incorporates several key innovations to enhance time series forecasting performance. One of its core features is adaptive lag selection, where the model learns which time lags are most relevant for a given forecasting task. This allows it to automatically focus on domain-specific lags—such as 24-hour, 7-day, or 12-

month intervals—that capture recurring temporal patterns and seasonal behaviours. In addition, Lag-Llama employs a multi-scale temporal representation strategy. It retains a dense representation of recent data to capture fine-grained trends while strategically sampling from the more distant past to model longer-term dependencies. The model uses logarithmic spacing for extremely longtime horizons to efficiently encode distant temporal information without overwhelming the model. Lag-LLama features a temporal embedding layer that maps each lag value into a high-dimensional space to support this rich temporal structure. It also uses positional encoding to preserve the relative timing of inputs. It incorporates calendar-based features like day-of-week, month, and holidays to capture recurring effects and seasonality better.

In this proposed methodology of monthly rainfall prediction using Lag-Llama, a sequential process transforms historical rainfall data into future forecasts using transformer-based attention mechanisms. Different context window sizes, attention patterns, and output aggregations are used for monthly predictions.

The two main modes of prediction in Lag-LLama are:

i)Zero-shot forecasting: Zero-shot forecasting is an emerging paradigm in time series forecasting. A pre-trained model can make accurate predictions on completely new, unseen time series without requiring additional training or fine-tuning on that specific data[11]. Even though the model has "zero" exposure to the target time series during training, it can still generate meaningful forecasts. Zero-shot forecasting is possible with Lag-LLama as it is trained on thousands of diverse time series from domains such as economic indicators, weather patterns, energy consumption, web traffic, and sales data. This wide range of training allows it to learn universal temporal patterns that generalise across different types of time series. Without additional training, this property will enable Lag- LLama to directly apply to new, unseen datasets, such as Kerala's rainfall. This forecasting approach provides immediate deployment capability and is a baseline for comparison with other fine-tuned models.

ii)Fine-Tuning Protocol: Fine-tuning involves updating the model's weights using historical data from Kerala's specific target domain's rainfall dataset. This process allows the model to adjust its parameters to fit better local trends, seasonality, and anomalies that may not be present in the pretraining data. Fine-tuning improves predictive accuracy, especially in regions or applications with unique characteristics or non-stationary dynamics. Fine-tuning can achieve high performance as the model starts from a strong pre-trained base.

Lag-Llama Architecture for Time Series Forecasting

i) Decoder-Only Transformer Backbone: Lag-LLama is built on a decoder-only transformer architecture, inspired by the LLama model used in language modelling [9]. This architecture is highly effective at modelling sequential data and capturing long-range dependencies, which are crucial for time series forecasting

ii) Input Representation: At each time step, the model receives:

- The current value of the univariate time series.
- A set of lagged values known as lag features.
- Covariate features, such as date-time information, here it is the month and year, which helps the model to understand temporal patterns and seasonality

iii) Embedding and Positional Encoding: Inputs are embedded and enhanced with Rotary Positional Encoding (RoPE)[12], allowing the model to distinguish between different positions in the sequence and efficiently capture periodicities and trends.

iv) Masked Decoder Layers: The model processes the input through multiple masked decoder layers, ensuring that predictions for a given time step only depend on current and past information

v) Distribution Head: Instead of giving a single value output, Lag-Llama predicts the parameters of a probability distribution (e.g., mean and variance for a Gaussian), enabling probabilistic forecasting. Figure 2 shows the Lag-Llama architecture [13]

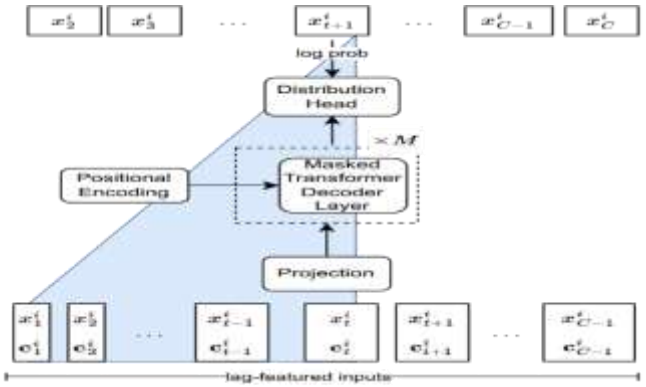

Figure. 2: Lag-Llama architecture

2.2 LSTM

LSTM networks have emerged as a robust deep learning architecture for sequential data modelling, particularly in time series forecasting applications. Unlike traditional recurrent neural networks (RNNs), LSTMs are specifically designed to address the vanishing gradient problem and can effectively capture long-term dependencies in sequential data[14][15]. Instead of neurons, LSTM networks consist of memory blocks linked through layers[15].A block has components that make it smarter than a classical neuron and includes a memory for recent sequences. It contains gates that manage the block's state and output. A block processes an input sequence, and each gate within a block uses sigmoid activation units to decide whether it is triggered, making the change of state and addition of information flowing through the block conditional. Figure 3 shows the architecture of an LSTM unit.

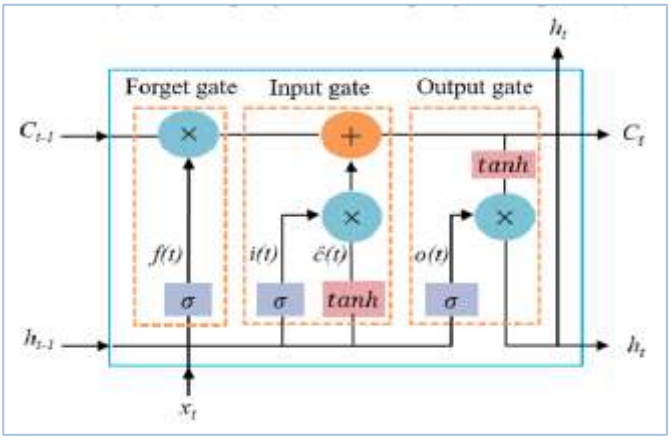

Figure 3: An LSTM unit architecture [14]

There are three types of gates within a unit [16]:

- **Forget Gate:** conditionally determines which information to discard from the block.

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f] \quad (1)$$

Where:

σ is the sigmoid function, which produces a value between 0 and 1. A value near 0 indicates "forget it almost completely," while a value near 1 indicates "retain it entirely.""

W_f is the weight matrix for the forget gate
h_{t-1} is the output from the previous time step

x_t is the current input

b_f is the bias term for the forget gate

- **Input Gate and Candidate Cell State**: The input gate determines which values will be updated in the cell state, while the candidate cell state represents a filtered version of the input data, prepared to be potentially added to the actual cell state.

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i] \quad (2)$$

$$\tilde{C}_t = tanh(W_c.[h_{t-1}, x_t] + b_c] \quad (3)$$

Where:

i_t is the input gate

$\tilde{C}_t$ is the filtered version of the input data

$tanh$ is the hyperbolic tangent function outputs values between -1 and 1. It helps regulate the network's non-linear characteristics

W_i *and* W_c are the weight matrices for the input gate and the candidate cell state, respectively

b_i and b_c are the biases for the input gate and the candidate cell state.

LSTM's use of gates regulates the updates and mitigates the risk of vanishing gradients by selectively updating the cell state. This controlled updating process ensures that important information is retained over more extended sequences, significantly enhancing the network's memory capabilities.

- **Cell State Update (C_t):** The cell state is an element-wise addition of the old state multiplied by the forget gate and the candidate state multiplied by the input gate.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (4)$$

Where:

C_{t-1} is the previous cell state.

- **Output Gate:** The output gate controls the parts of the cell state that are output to the next layer or used in the final prediction.

$$O_t = \sigma(W_o.[h_{t-1}, x_t] + b_o \quad (5)$$
$$h_t = o_t * \tanh(C_t) \quad (6)$$

Where:

W_o is the weight matrix for the output gate

b_o is the bias.

Each unit is like a mini-state machine, with gates that have weights learned during the training procedure. LSTM's use of gates regulates the updates and mitigates the risk of vanishing gradients by selectively updating the cell state. This controlled updating process ensures that important information is retained over more extended sequences, significantly enhancing the network's memory capabilities.

In univariate forecasting, only past values of the rainfall series are used for prediction. The task is to predict future rainfall values solely based on past observations. Unlike deterministic forecasting, probabilistic forecasting offers a full predictive distribution, enabling the estimation of prediction intervals and uncertainty.

Dataset

Statistical analysis of the rainfall dataset indicates that Kerala receives an average annual rainfall of approximately 2,828 mm. The standard deviation of about 401 mm reflects a moderate level of inter-annual variability, with a coefficient of variation around 14%, suggesting relatively stable long-term rainfall patterns. Seasonal variation is strongly marked, with January recording the lowest mean monthly rainfall of about 11 mm and exhibiting the highest inter-annual variability. Fig. 4 illustrates the rainfall distribution across the entire period, with the training, validation, and testing periods distinguished by different colours.

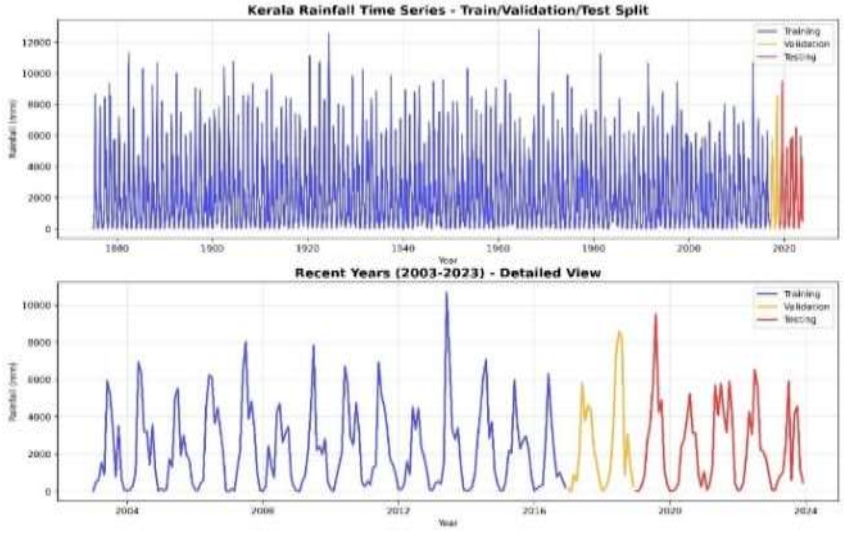

Figure 4: Rainfall time series data

The performance of Lag-Llama is compared against baseline models, incorporating both traditional statistical approaches and deep learning architectures. A deep learning model, the LSTM network, represents sequence modelling architectures capable of learning long-range temporal dependencies and nonlinearities in data. Next, a ML-based regression model, the Random Forest (RF), is also included to evaluate the model's comparative strength against ensemble-based learning algorithms that perform well on tabular data but lack sequential modelling capabilities. The study highlights its robustness across various paradigms by benchmarking Lag-LLama's performance in zero-shot and fine-tuned configurations against LSTM and Random Forest. In this work, all the prediction models, Lag-LLama, LSTM, and Random Forest, are evaluated in a univariate forecasting setup, where each time series is modelled and predicted independently. While Lag-LLama inherently produces probabilistic forecasting, the other models—LSTM and Random Forest—were initially developed for point forecasts. To ensure a fair comparison, we adapt these baseline models for probabilistic prediction by incorporating distributional output layers and applying key enhancements inspired by Lag-LLama, such as value scaling and temporal feature embedding (e.g., month/season).

The performance of each model is evaluated using consistent metrics, including the Continuous Ranked Probability Score (CRPS) [17] for probabilistic accuracy, whereas Mean Absolute Error(MAE) and Root Mean Squared Error (RMSE) for point prediction performance. The Continuous Ranked Probability Score (CRPS) is a proper scoring rule for evaluating probabilistic forecasts against observed outcomes. Mathematically, it is defined as:

$$CRPS(F, y) = \int_{-\infty}^{\infty} [F(x) - 1\{x \geq y\}]^2 \, dx \qquad (7)$$

- $F(x)$ is the cumulative distribution function (CDF) of the forecast
- y is the observed (true) value
- $1\{x \geq y\}$ is the indicator function (Heaviside step function) that equals 1 if $x \geq 1$ and 0 otherwise.

3.Results And Discussions

A strict timestamp-based non-overlapping split for training, validation, and testing is adapted to ensure robust evaluation and prevent data leakage in time series forecasting. The dataset used in this study comprises monthly rainfall data for Kerala, India, spanning from January 1871 to December 2023. The dataset is split chronologically as follows:

- Training Set: January 1871 to December 2000
- Validation Set: January 2001 to December 2013
- Test Set: January 2014 to December 2023

During pretraining, the model is trained exclusively on the training set. At the end of each epoch, the model's performance is evaluated on the internal validation windows. The average validation loss over these windows is computed and used as the early stopping criterion. For the fine-tuning phase, as well as for all supervised baselines (e.g., LSTM and Random Forest), we adopt the same structured splitting. In this study, we fine-tune the Lag-LLaMA model to capture the temporal dynamics of rainfall data. The model's performance is susceptible to key hyperparameters such as context length, learning rate, batch size, number of attention heads, model dimension, and dropout rates. These parameters directly influence the model's ability to learn from historical rainfall patterns while generalising effectively to unseen data.

A random search across different hyperparameters is done. The optimal configuration is selected based on validation loss computed on our Kerala rainfall dataset spanning 1871-2023. The hyperparameters selected based on the optimal CRPS values are presented in Table1. The hyperparameters are chosen from a range of values using hyperparameter search.

Table 1: Hyperparameters chosen for the Lag-Llama Model

Hyperparameter	Value Range	Value Selected
No. of Layers	{1-9}	4
No. of Heads	{1-9}	5

Embedding Dimensions per Head	{16,32,64,128,256,512}	32
Context Length	{32,64,128,256,512,1024}	48
Learning Rate	{$1e^{-4}$ to $10e^{-4}$}	5e-4

The zero-shot performance of the Lag-Llama model on Kerala monthly rainfall data, where no rainfall data are used for fine-tuning, is evaluated. The Lag-Llama model is then fine-tuned using the Kerala rainfall dataset (1871-2023), and the model performance is evaluated. This approach allows for studying how a pre-trained foundation model adapts to the unique seasonal patterns, monsoon dynamics, and long-term climate variability inherent in Kerala's rainfall data when substantial historical observations spanning over 150 years are available for domain-specific training.

Figure 5 depicts the predicted monthly rainfall plots for Kerala from 2014 to 2023, showing the probabilistic forecasting results with mean predictions, median predictions, and 80% confidence intervals (shaded region). The confidence intervals represent the uncertainty bounds of the rainfall predictions, capturing the inherent variability and seasonal patterns in Kerala's monsoon-driven precipitation regime over the 10-year evaluation period.

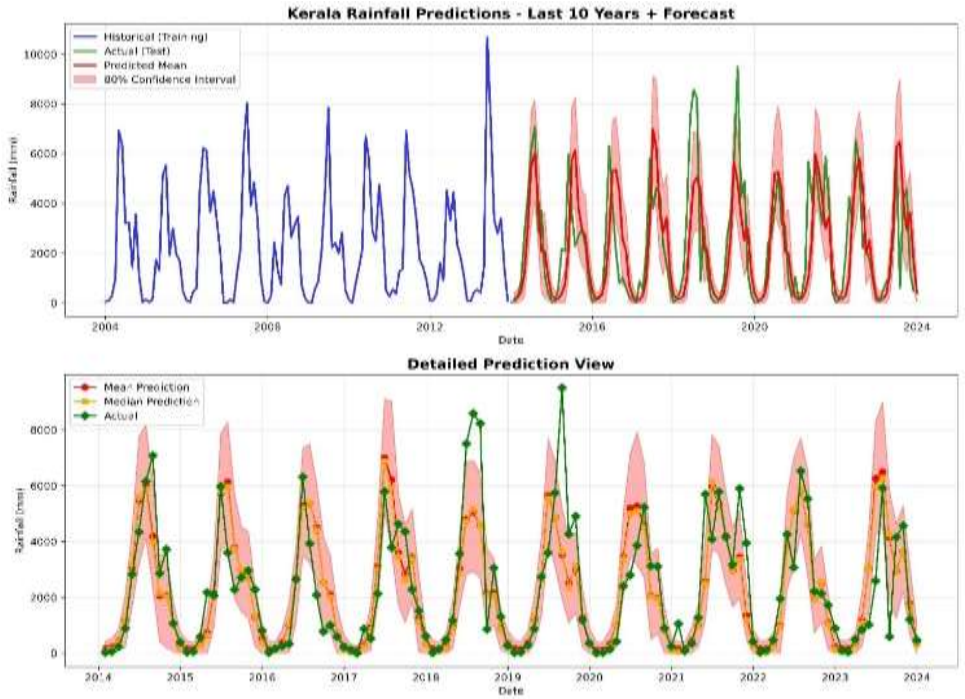

Figure 5: Monthly Rainfall Forecasts for Kerala (2014-2023) using Lag-Llama: Mean, Median, and 80% Confidence Intervals

LSTM is selected as the deep learning baseline model for performance comparison against Lag-Llama in monthly rainfall forecasting tasks. LSTM networks are a class of RNNs effective for capturing long-term dependencies in time series data. To generate probabilistic forecasts rather than point forecast values, the LSTM output is treated as a distribution parameterised to capture uncertainty, which allows for the computation of forecast

intervals and evaluation using CRPS. The model was trained on the monthly rainfall data from 1871 to 2000, with the last 10 years (2014–2023) used for testing. The key hyperparameter values used are listed in Table 2. Here also hyperparameters are selected through search from a range of values.

Table 2: Hyperparameters used for LSTM

Hyperparameter	**Value Selected**

Input sequence length	48 months
Hidden Size	128
No. of LSTM layers	2
Dropout	0.3
Batch Size	64
Learning rate	0.0001
Optimizer	Adam
Epochs	100

Figure 6 depicts Kerala's predicted monthly rainfall plots using LSTM from 2014 to 2023, showing the probabilistic forecasting results with mean predictions and 80% confidence intervals (shaded region).

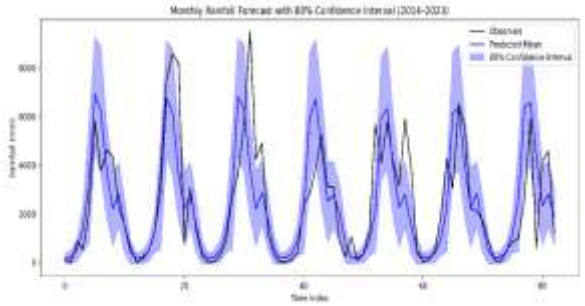

Figure 6: Monthly Rainfall Forecasts for Kerala (2014-2023) using LSTM: Mean and 80% Confidence Intervals

The Random Forest (RF) regression model is robust in handling nonlinear relationships and feature interactions. For probabilistic forecasting, quantile regression forests are used to estimate conditional quantiles of the rainfall distribution instead of single-point estimates. This enables the construction of prediction intervals (e.g., 80% confidence intervals) that capture the uncertainty in rainfall forecasts. The RF model was trained on historical data from 1871 to 2000, validated on 2001–2013, and tested on 2014–2023. The number of trees or estimators used is 200. Figure 7 depicts Kerala's predicted monthly rainfall plots using RF from 2014 to 2023, showing the probabilistic forecasting results with mean predictions and 80% confidence intervals (shaded region).

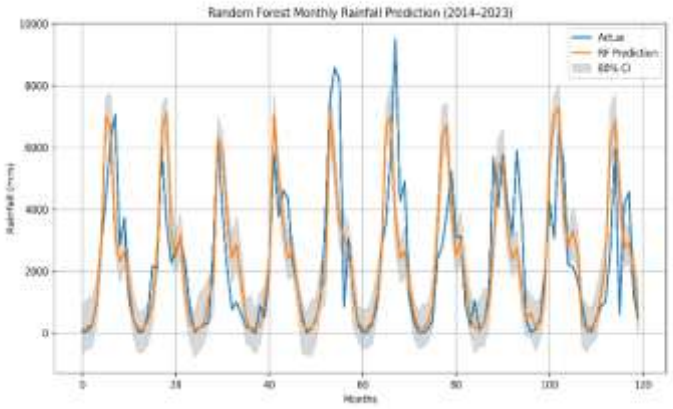

Figure 7: Monthly Rainfall Forecasts for Kerala (2014-2023) using RF: Mean and 80% Confidence Intervals

Table 3 is the performance comparison table for monthly rainfall prediction using three models: Random Forest (RF), LSTM, and Lag-LLaMA (both zero-shot and fine-tuned). The table compares four evaluation metrics:

- Normalised CRPS – assesses probabilistic forecast accuracy (lower is better)
- MAE – Mean Absolute Error (lower is better)
- MSE – Mean Squared Error (lower is better)
- RMSE – Root Mean Squared Error (lower is better)

Table 3: Performance comparison table for monthly rainfall prediction

Model	Normalized CRPS	MAE(mm)	MSE(mm^2)	RMSE(mm)
Random Forest	0.36	970	2268460	1506
LSTM	0.41	1006	1987106	1409
Lag-LLama(zero-shot)	0.36	1034	2551079	1597
Lag-LLama(Fine-tuned)	0.21	595	883781	940

The comparative evaluation of four forecasting models for monthly rainfall prediction in Kerala reveals significant differences in performance across key metrics, with the fine-tuned Lag-LLaMA model emerging as the most accurate. This model achieved the best results across all evaluation criteria, recording a normalised CRPS of 0.21, MAE of 595 mm, MSE of 883,781 mm^2, and RMSE of 940 mm—demonstrating its strong ability to capture the temporal dynamics of Kerala's rainfall patterns.

In contrast, traditional models such as Random Forest and zero-shot Lag-LLaMA showed moderate performance, recording a normalised CRPS of 0.36. However, Random Forest slightly outperformed zero-shot Lag-LLaMA in point forecasting, achieving lower MAE (970 mm vs. 1034 mm) and RMSE (1506 mm vs. 1597 mm). The LSTM model exhibited the weakest performance overall, with the highest normalised CRPS (0.41) and an MAE of 1006 mm, although its MSE and RMSE remained comparable to zero-shot Lag-LLaMA. Notably, the substantial improvement from zero-shot to fine-tuned Lag-LLaMA highlights the importance of domain-specific training: fine-tuning reduced RMSE by approximately 660 mm and improved normalised CRPS by 0.15.

These results emphasise that while pretrained foundation models like Lag-LLaMA offer strong baseline capabilities, fine-tuning on local data is essential for achieving high accuracy in real-world, domain-specific time series forecasting tasks such as Kerala's rainfall prediction.

4.Conclusion

This work has demonstrated that while traditional ML algorithms like Random Forest and deep learning approaches like LSTM offer significant improvements over classical methods, the emergence of foundation models marks a paradigm shift in time series forecasting capabilities. The superior performance of the fine-tuned Lag-LLama model, achieving 41-42% improvements in prediction accuracy over zero-shot performance and surpassing both Random Forest and LSTM approaches, establishes a new benchmark for rainfall forecasting in Kerala. The successful application of Lag-LLama to Kerala rainfall forecasting establishes a methodology that can be adapted to other regions with similar climatic characteristics. As Kerala continues to face the challenges of climate variability and change, the advanced forecasting capabilities developed through this research offer essential tools for strengthening resilience and adaptive capacity. The study demonstrates the critical importance of domain-specific fine-tuning for foundation models. Foundation models are continuously evolving, and innovations are being developed. This innovative application of Lag-LLama for rainfall forecasting will provide a strong foundation for advanced meteorological applications.

References

1. Chattopadhyay, S., & Chattopadhyay, G. (2010). Univariate modelling of summer-monsoon rainfall time series: Comparison between ARIMA and ARNN. Comptes Rendus Geoscience, 342(2), 100–107. https://doi.org/10.1016/j.crte.2009.10.016
2. R. Kamath and R. Kamat, "Time-series Analysis and Forecasting of Rainfall at Idukki district, Kerala: Machine Learning Approach," Disaster Advances, vol. 11, pp. 27–33, 2018.
3. O. A. Wani, S. S. Mahdi, M. Yeasin et al., "Predicting rainfall using machine learning, deep learning, and time series models across an altitudinal gradient in the North-Western Himalayas," Scientific Reports, vol. 14, p. 27876, 2024. doi: 10.1038/s41598-024-77687-x

4.]V. Kumar, N. Kedam, O. Kisi, et al., “A comparative study of machine learning models for daily and weekly rainfall forecasting,” Water Resources Management, vol. 39, pp. 271–290, 2025, doi: 10.1007/s11269-024-03969-8.
5. Jeyaprakash, D. Harsha, D. Chowdary, B. Kumaar, G. Rahul, S. Vishvanathan, E. A. Gopalakrishnan, and M. Dhanya, "Going Beyond Traditional Methods: Using LSTM Networks to Predict Rainfall in Kerala," pp. 112–121, 2024, doi: 10.1007/978-3-031-53717-2_11.
6. J. Saikia, K. Goswami, and S. C. Kakaty, "Wavelet-SARIMA-Transformer: A Hybrid Model for Rainfall Forecasting," arXiv, 2025, doi: 10.48550/arXiv.2509.11903.
7. J. Adhikary and R. Maiti, "Long-Term Spatio-Temporal Forecasting of Monthly Rainfall in West Bengal Using Ensemble Learning Approaches," arXiv preprint arXiv:2510.13927, 2025. [Online]. Available: https://arxiv.org/abs/2510.13927
8. D.C.R.Novitasari, B.D. Supatmanto, M.F. Rozi, Hermansah, Y. Farida, “ Rainfall Prediction Based on Himawari-8 IR Enhanced Image Using Backpropagation,” Journal of Physics: Conference Series; Bristol Vol. 1501, Iss. 1, (Mar 2020). DOI: 10.1088/1742-6596/1501/1/012011.
9. A. Goyal, Y. Qin, and L. Yang, "Lag-Llama: Towards Foundation Models for Probabilistic Time Series Forecasting," in Proc. 41st Int. Conf. Machine Learning, 2024, pp. 3428-3437.
10. K. Rasul, A. Ashok, A. R. Williams, H. Ghonia, et al., "Lag-Llama: Towards Foundation Models for Probabilistic Time Series Forecasting," arXiv preprint arXiv:2310.08278, Oct. 2023.
11. N. Gruver, M. Finzi, S. Qiu, and A. G. Wilson, "Large Language Models Are Zero-Shot Time Series Forecasters," in Advances in Neural Information Processing Systems, vol. 36, pp. 1–13, 2024.
12. J. Su, Y. Lu, S. Pan, A. Murtadha, B. Wen, and Y. Liu, "RoFormer: Enhanced Transformer with Rotary Position Embedding," Neurocomputing, vol. 568, p. 127063, 2024.
13. K. Rasul, C. Seward, I. Schleich, and R. Vollgraf, "Autoregressive Denoising Diffusion Models for Multivariate Probabilistic Time Series Forecasting," in Proc. 40th Int. Conf. Machine Learning, 2023, pp. 28752-28767.
14. Y. Bengio, P. Simard, and P. Frasconi, "Learning Long-Term Dependencies with Gradient Descent is Difficult," IEEE Transactions on Neural Networks, vol. 5, no. 2, pp. 157–166, 1994.
15. S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
16. M. Mohsen, A. Saeed, and S. G. Elkaseer, "Industry 4.0-Oriented Deep Learning Models for Human Activity Recognition," IEEE Access, vol. 9, pp. 1-1, 2021, doi: 10.1109/ACCESS.2021.3125733.
17. F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," Neural Computation, vol. 12, no. 10, pp. 2451-2471, 2000.