Cherrapunji Rainfall Forecasting using Time Series Data

A Project Report Submitted

for the Course

MA498 Project I

by

RAJ BOROGAON

(Roll No. 210123049)

LALHRIEMSANG FAIHRIEM

(Roll No. 210123036)

to the

DEPARTMENT OF MATHEMATICS

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

GUWAHATI - 781039, INDIA

November 2024

CERTIFICATE

This is to certify that the work contained in this project report entitled "Title of the project report" submitted by RAJ BOROGAON (Roll No.: 210123049) and LALHRIEMSANG FAIHRIEM (Roll No.: 210123036) to the Department of Mathematics, Indian Institute of Technology Guwahati towards partial requirement of Bachelor of Technology in Mathematics and Computing has been carried out by him/her under my supervision.

It is also certified that, along with literature survey, a few new results are established/computational implementations have been carried out/simulation studies have been carried out/empirical analysis has been done by the student under the project.

Turnitin Similarity: __ %

Guwahati - 781 039 (Prof. Bhupen Deka)

November 2024 Project Supervisor

ABSTRACT

The main aim of the project is analysing the rainfall data available and forecasting rainfall for next years.

Contents

List of Figures

Chapter 1

Introduction

Rainfall is a crucial environmental factor that affects ecology, agriculture, and socio-economic conditions. The traditional use of machine learning and deep learning algorithms for predicting rainfall plays a vital role in managing these impacts, enhancing preparedness for extreme weather events, optimizing resource allocation, and saving lives while reducing economic losses. This report aims to examine the factors influencing rainfall patterns, evaluate current forecasting methods, and explore different strategies to improve prediction accuracy.

1.1 Background

The North Eastern regions of India, particularly the Cherrapunjee district, have experienced significant rainfall with extreme variations over the years, making rainfall prediction challenging. This report focuses on Cherrapunjee's historical rainfall data—known for receiving the highest precipitation in the world—analyzing the data, and exploring different strategies for implementing machine learning and deep learning models to improve prediction

accuracy for the region.

1.2 Problem Statement

Given a historical time series rainfall data of Cherapunjee district, we are tasked with finding and implementing suitable machine learning / deep learning technique to forecast the rainfall in Cherapubjee and also looking at the trends to draw conclusion of model performances.

1.3 Scope

Using the results we obtained, we can conclude the probable rainfall patterns for the coming years. This information can be used to make informed decisions and adjust policies, such as implementing precautionary measures for extreme weather events, optimizing agricultural planning, adjusting planting schedules, and managing water resources to boost agricultural yields. Additionally, sectors like energy production can plan for hydroelectric power generation based on expected rainfall, while urban planning can improve flood management and infrastructure development to minimize weather-related disruptions.

Chapter 2

Theoretical Background

In this sections the methods used for time series forecasting and evaluation metrics are listed.

2.1 Linear Regression

Linear Regression models the relationship between the target variable y and the predictor variable x (or multiple predictors in multiple regression) by fitting a straight line to the data.

$$y = \beta_0 + \beta_1 x + \epsilon$$

where:

- y is the target variable,
- x is the predictor variable,
- β_0 is the intercept,
- β_1 is the slope (coefficient),

- ϵ is the error term.

Linear regression assumes a linear relationship between variables, making it suitable for simple datasets with no complex interactions.

2.2 Random Forest Regression

Random Forest Regression is an ensemble method that builds multiple decision trees and averages their predictions. It is useful for handling complex, non-linear relationships.

The prediction for a new data point x is the average of the predictions of all trees:

$$\hat{y} = \frac{1}{N} \sum_{i=1}^{N} f_i(x)$$

where:

- $\hat{y}$ is the predicted value,
- N is the number of trees in the forest,
- $f_i(x)$ is the prediction from the i^{th} tree.

Random Forest is particularly powerful for capturing non-linear patterns and interactions in the data.

2.3 Time Series Data

Many statistical methods relate to data which are independent, or at least uncorrelated. There are many practical situations where data might be correlated. This is particularly so where repeated observations on a given system

are made sequentially in time. Data gathered sequentially in time are called a time series.

Examples

Here are some examples in which time series arise:

- Economics and Finance
- Environmental Modelling
- Engineering

The simplest form of data is a long-ish series of continuous measurements at equally spaced time points, such as

- observations are made at distinct points in time, these time points being equally spaced
- and, the observations may take values from a continuous distribution.

2.3.1 Covariance and Correlation

Covariance is defined as the variance of two variables. To calculate covariance between x and y or x1 and x2 below formula can be used

$$\mathrm{Cov}(X,Y) = \frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y}) \tag{2.1}$$

Correlation defined as how much increase(decrease) in one variable causes other variable to increase(decrease) or vice-versa. E.g.(Salary increases with experience). And it can be calculated as

$$r_{XY} = \frac{\mathrm{Cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{\frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^2} \cdot \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \tag{2.2}$$

2.3.2 Auto Covariance

Auto Covariance is defined as the covariance between the present value x_t with previous value x_{t-1} and the present value x_t with x_{t-2}. So the formula will become

$$\frac{1}{n}\sum_{i=1}^{n}(X_i - \mu)(X_{t-1} - \mu) \tag{2.3}$$

2.3.3 Auto Correlation (ACF)

In time series we deal with variables with respect to time like stock price, sales of a company over the years, temperature of a area etc. So while predicting the future value past values might be useful, we use Autocorrelation(ACF) function to measure the correlation of a variable with its own past values, we analyze time series data to understand how a time series is related to it's previous values.

$$\rho_k = \frac{\sum_{t=k+1}^{n}(X_t - \bar{X})(X_{t-k} - \bar{X})}{\sum_{t=1}^{n}(X_t - \bar{X})^2} \tag{2.4}$$

ρ_k : partial autocorrelation at lag k,

2.3.4 Partial Auto Correlation (PACF)

In autocorrelation we find correlation between present(x_t) and next (x_t-1) values. In Partial Autocorrelation is finding the correlation between present x_t random lags value (x_t-h) so, the correlation in the middle values like $(x_t - 1)(x_t - 2)(x_t - 3)\ldots.(x_t - (h - 1))$ will not be taken into account.

The partial autocorrelation at lag k is the correlation that results after removing the effect of any correlations due to the terms at shorter lags.

$$\phi_k = \frac{\mathrm{Cov}(X_t, X_{t-k} \mid X_{t-1}, X_{t-2}, \ldots, X_{t-(k-1)})}{\sqrt{\mathrm{Var}(X_t \mid X_{t-1}, X_{t-2}, \ldots, X_{t-(k-1)}) \cdot \mathrm{Var}(X_{t-k} \mid X_{t-1}, X_{t-2}, \ldots, X_{t-(k-1)})}} \tag{2.5}$$

2.3.5 Stationarity Time series data

A Time Series is stationary if it satisfies the following conditions:

- Constant μ (mean) for all t.
- Constant σ (variance) for all t.
- The autocovariance function between X_{t_1} and X_{t_2} only depends on the interval t_1 and t_2

2.3.6 Augmented Ducky Fuller (ADF) Test

Augmented Ducky Fuller test is a statistical test used to test whether a given Time Series is stationary or not. The primary purpose of the ADF test is to test for the existance of unit root in the time series. Non-stationary series can lead to unreliable statistical inference and forecast models. Therefore, detecting stationarity is a key step in time series analysis.

Unit Root of Time Series and Random Walk

The presence of a unit root in a time series means that the series is non-stationary and its statistical properties (mean, variance, and autocorrelation) change over time. A common feature of such series is that shocks or disturbances to the system have a permanent effect, and the series does not "return" to its previous level or mean after the shock has occurred.

consider the equation

$$X_t = X_{t-1} + \epsilon_t \tag{2.6}$$

This is called a random walk model, where the current value of the time series is simply the previous value plus a random shock. In this case, the time series has a unit root because the coefficient of the lagged value X_{t-1} is equal to 1.

Hypothesis of the ADF Test

The null hypothesis H_0 and alternate hypothesis H_1 of the ADF test are as follows:

- H_0: The time series has a unit root, meaning it is non-stationary.
- H_1: The time series does not have a unit root, meaning it is stationary.

The augmented Dicky-Fuller Test Equation

The ADF test is based on the following regression equation:

$$\Delta X_t = \alpha + \beta t + \gamma X_{t-1} + \sum_{i=1}^{p} \delta_i \Delta X_{t-i} + \epsilon_t \tag{2.7}$$

where:

- X_t is the time series at time t,
- $\Delta\ X_t = X_t$ - X_{t-1} is the first difference of the series,
- α is a constant (drift) term,
- β t is a deterministic trend component (optional depending on the trend assumption),

- γ is the coefficient of the lagged value of the series (X_{t-1}),
- δ_i are the coefficients of the lagged difference terms ($\Delta\ X_{t-i}$)
- p is the number of lags included to account for autocorrelation
- ϵ_t is the error term

Fit the regression equation on the time series data. The ADF test statistic computed and compared with the critical values (usually from a table or statistical software). If the test statistic is less than the critical value, we reject the null hypothesis and conclude that the series is stationary.

2.4 Auto Regression(AR) Model

An AutoRegressive (AR) model is a type of statistical model used in time series analysis, where the current value of a series is regressed on its own past values. In an AR model, the future value of the time series is assumed to depend linearly on previous values.

The equation for an AutoRegressive model of order p (AR(p)) is given by:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + \epsilon_t \tag{2.8}$$

where:

- X_t is the value of the time series at time t,
- $\phi_1, \phi_2, \ldots, \phi_p$ are the autoregressive coefficients,
- ϵ_t is the white noise error term at time t.

2.5 Moving Average(MA) Model

A Moving Average (MA) model is a time series model that expresses the current value of a series as a linear function of past white noise error terms (random shocks). In an MA(q) model, the current value of the series is modeled as a function of the current and previous q error terms. Moving Average models are particularly useful for capturing short-term correlations in data and are often combined with AR models in ARMA and ARIMA frameworks.

The equation for a Moving Average model of order q (MA(q)) is given by:

$$X_t = \mu + \epsilon_t + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2} + \cdots + \theta_q\epsilon_{t-q} \tag{2.9}$$

where:

- X_t is the value of the time series at time t,
- μ is the mean of the series,
- ϵ_t is the white noise error term at time t,
- $\theta_1, \theta_2, \ldots, \theta_q$ are the moving average coefficients.

2.6 ARMA, ARIMA, SARIMA, SARIMAX

Auto Regressive Moving Average (ARMA) Model

The ARMA model is a combination of the AR and MA models. It is used for stationary time series data and provides a way to capture both short-term and long-term correlations in a single model

ARMA(p, q) model has two components

- AR(p): The Auto Regressive Part, which regresses the current value on p previous values.
- MA(q): The Moving Average part, which regresses the current value on q previous error terms.

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \epsilon_{t-q} \quad (2.10)$$

where:

- X_t is the value of the time series at time t,
- $\phi_1, \phi_2, \ldots, \phi_p$ are the autoregressive coefficients,
- $\theta_1, \theta_2, \ldots, \theta_q$ are the moving average coefficients,
- ϵ_t is the white noise error term.

Auto Regressive Moving Average (ARMA) Model

The ARIMA model extends the ARMA model by including differencing to make non-stationary data stationary. This model is widely used for time series that exhibit trends or patterns over time.

ARIMA(p,d,q) model has two components

- AR(p): The Auto Regressive Part, which uses p past values.
- I(d): The integrated part, which represents the order of differencing needed to make the series stationary.
- MA(q): The Moving Average part, which uses q past terms

The equation for an ARIMA(p, d, q) model is:

$$(1 - \sum_{i=1}^{p} \phi_i L^i)(1 - L)^d X_t = \theta_0 + (1 + \sum_{j=1}^{q} \theta_j L^j)\epsilon_t \quad (2.11)$$

where:

- L is the lag operator,
- $(1 - L)^d$ represents differencing to achieve stationarity.

Seasonal AutoRegressive Integrated Moving Average (SARIMA)

SARIMA extends the ARIMA model by incorporating seasonal components to handle data with periodic fluctuations or seasonality (e.g., monthly sales data that peaks every December). SARIMA is often denoted as SARIMA(p, d, q)(P, D, Q, s), where (p,d,q) are the non-seasonal parameters,(P,D,Q) are the seasonal parameters, and s is the seasonal period (e.g. s=12 for monthly data with yearly seasonality).

Parameters of SARIMA

- Non seasonal parameters (p,d,q):
  - p: Number of autoregressive terms.
  - d: Number of differences required to make the series stationary.
  - q: Number of moving average terms.
- Seasonal parameters(P,D,Q,s):
  - P: Seasonal autoregressive terms.
  - D: Seasonal differencing to remove seasonal trends.
  - Q: Seasonal moving average terms.
  - s: Seasonal period

The equation for a SARIMA model of order $(p, d, q)(P, D, Q, s)$ is given by:

$$\Phi_P(L^s)(1 - L^s)^D(1 - L)^d X_t = \Theta_Q(L^s)\epsilon_t + \theta(L)\epsilon_t \tag{2.12}$$

where:

- $\Phi_P(L^s)$ is the seasonal autoregressive operator of order P,
- $(1 - L^s)^D$ is the seasonal differencing operator,
- $\Theta_Q(L^s)$ is the seasonal moving average operator of order Q,
- s is the seasonal period.

Seasonal AutoRegressive Integrated Moving Average with Exogenous Variables (SARIMAX)

SARIMAX is an extension of SARIMA that includes exogenous variables (often denoted as X_t), which are external factors or predictors that may influence the target variable. These exogenous variables can help improve forecasting accuracy when there are known factors that impact the time series.

For example, while forecasting monthly sales, you may include promotional spending or holiday seasons as exogenous variables that affect the sales.

The equation for a SARIMAX model, which includes exogenous variables X_t, is:

$$\Phi_P(L^s)(1 - L^s)^D(1 - L)^d X_t = \Theta_Q(L^s)\epsilon_t + \theta(L)\epsilon_t + \beta X_t \tag{2.13}$$

where:

- X_t represents the exogenous variables at time t,
- β is a vector of coefficients for the exogenous variables.

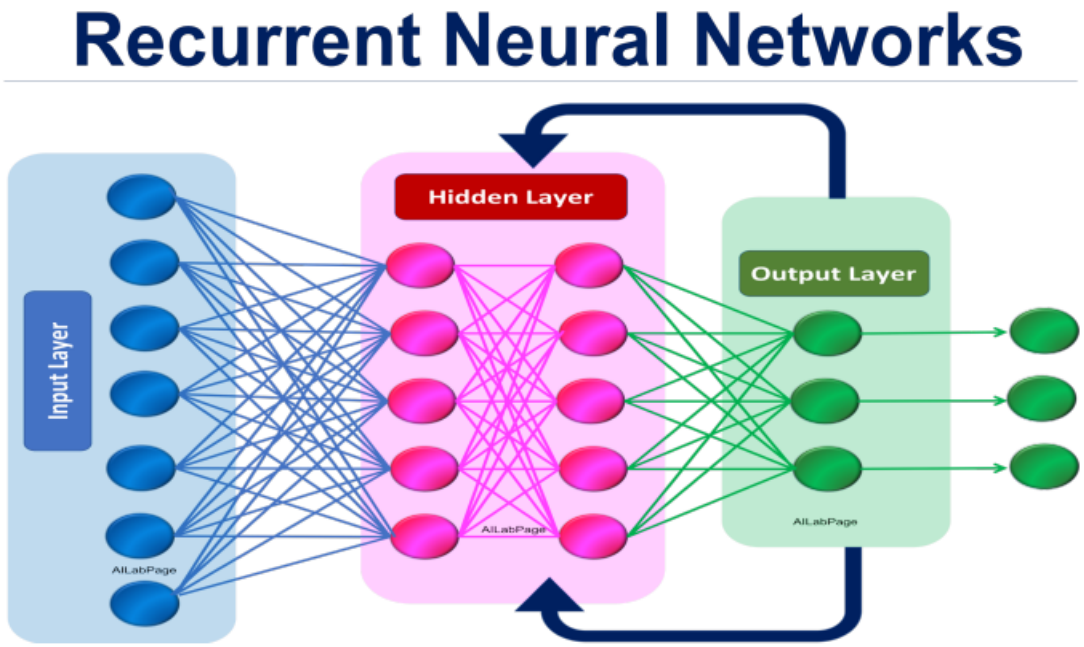


Figure 2.1: RNN

2.7 Recurrent Neural Network (RNN)

RNNs (Recurrent Neural Networks) are a class of neural networks which are powerful and used for modelling sequence data such as time series or natural language. RNNs use a for loop which iterates over the time steps of a sequence while maintaining an internal state which encodes information about the time steps it has gone over. This means that the output layer can record the data it receives and send it back into the input layer to recalculate based on the previous findings. This is illustrated in Figure 2.1.

2.7.1 Long Short-Term Memory (LSTM)

LSTM is a type of recurrent neural network (RNN) designed to capture long-term dependencies in time series data. It overcomes the vanishing gradient

problem that traditional RNNs face, making it effective for forecasting sequences with long-term dependencies.

The LSTM model learns the relationships between past observations and the target variable, adapting its weights over time to minimize forecasting errors.

The equation for an LSTM network involves the following components:

$$\begin{aligned} f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) && \text{(forget gate)} \\ i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) && \text{(input gate)} \\ \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) && \text{(candidate memory)} \\ C_t &= f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t && \text{(cell state)} \\ o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) && \text{(output gate)} \\ h_t &= o_t \cdot \tanh(C_t) && \text{(hidden state)} \end{aligned} \tag{2.14}$$

where:

- x_t represents the input at time t,
- h_t is the hidden state at time t,
- C_t is the cell state at time t,
- f_t, i_t, o_t are the forget, input, and output gates, respectively,
- W_f, W_i, W_C, W_o and b_f, b_i, b_C, b_o are the weights and biases associated with each gate.

2.8 RMSE, MAE, MAPE, R2 score

In time series and regression analysis, RMSE, MAE, MAPE, and R2 score are commonly used metrics to evaluate model performance. Each metric assesses

prediction accuracy from a different angle and is applicable to various models, such as ARIMA, SARIMA, SARIMAX, and linear regression models.

Root Mean Squared Error (RMSE)

RMSE measures the average magnitude of the error (difference) between observed and predicted values. It gives higher weight to large errors, making it useful for applications where large errors are particularly undesirable.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2.15}$$

Mean Absolute Error (MAE)

MAE calculates the average of the absolute differences between observed and predicted values. Unlike RMSE, it does not give additional weight to large errors, making it a straightforward measure of average error size.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2.16}$$

Mean Absolute Percentage Error (MAPE)

MAPE expresses the average error as a percentage of the actual values. This metric is useful when you want error to be scaled relative to the magnitude of the actual values, although it can be misleading with values close to zero.

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100 \tag{2.17}$$

Coefficient of Determination (R^2 Score)

R^2 score measures the proportion of variance in the observed data that is predictable from the model. It ranges from 0 to 1, with values closer to 1 indicating a better fit. Negative values can occur when the model performs worse than a simple average.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{2.18}$$

where:

- y_i is the actual value,
- $\hat{y}_i$ is the predicted value,
- $\bar{y}$ is the mean of the actual values,
- n is the number of observations.

Chapter 3

Data Analysis

In this section we plot the monthly wise precipitation and overall precipitation to understand how data behaved over the years.

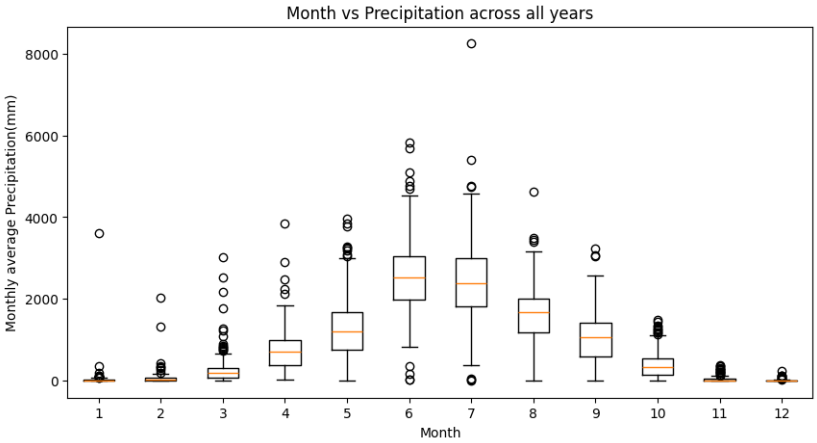
We can observe some interesting things which are generally not expected. In the fig 3.2 the graph shows that though in January rainfall is all time low tho around 1961-1962 rainfall was as high as around June and July. Similarly in Fig 3.3, around 2000-2001 rainfall in February was as high as June and July.

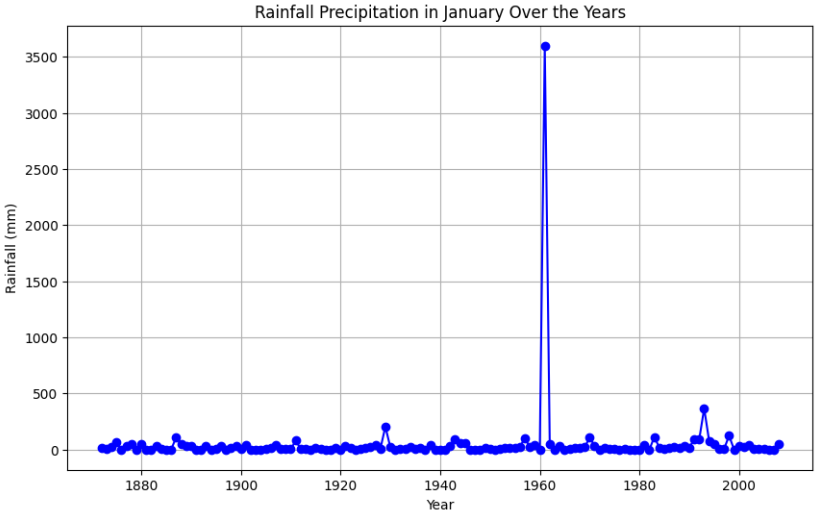
In fig 3.6, rainfall for May is plotted and though rainfall for May is not expected to be that low, it was very low for some period around 1990.

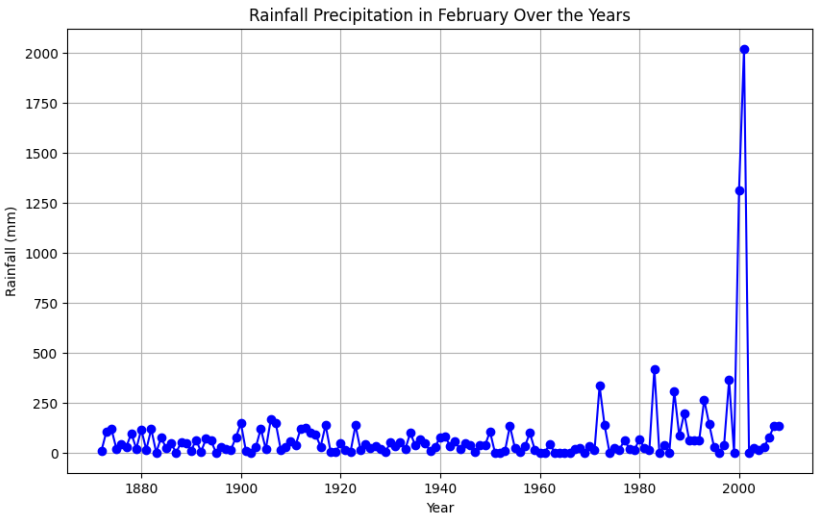
In fig 3.8, rainfall for July is plotted and we can see that rainfall for July is not that generally now but during 1962 the rainfall was very low and around 1970-1972 it went very high.

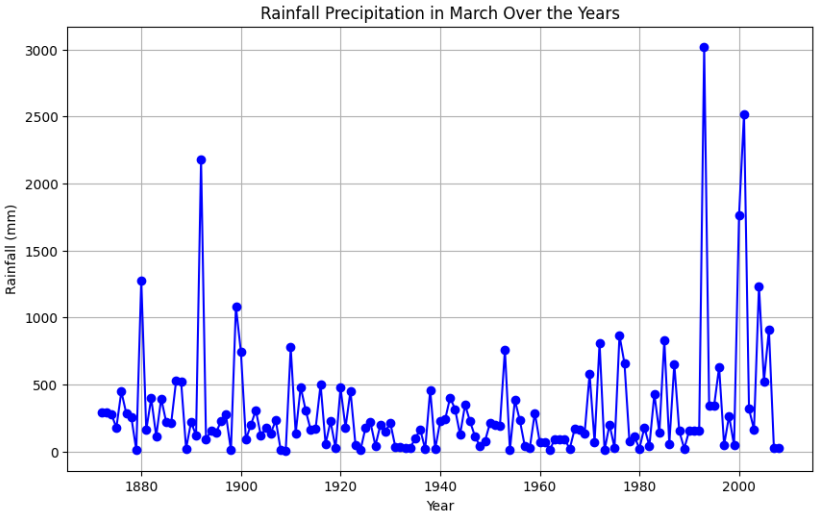
In fig 3.11, rainfall for October is plotted and we can see that rainfall for October is sometimes very high and sometimes very low and it went very low for sometime in 1990s.

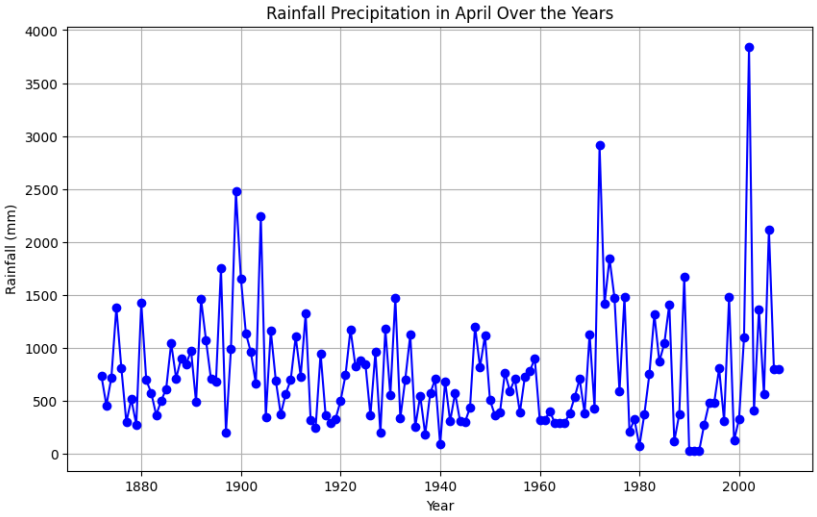
Similarly we can observe many outliers and general trend from the plots.

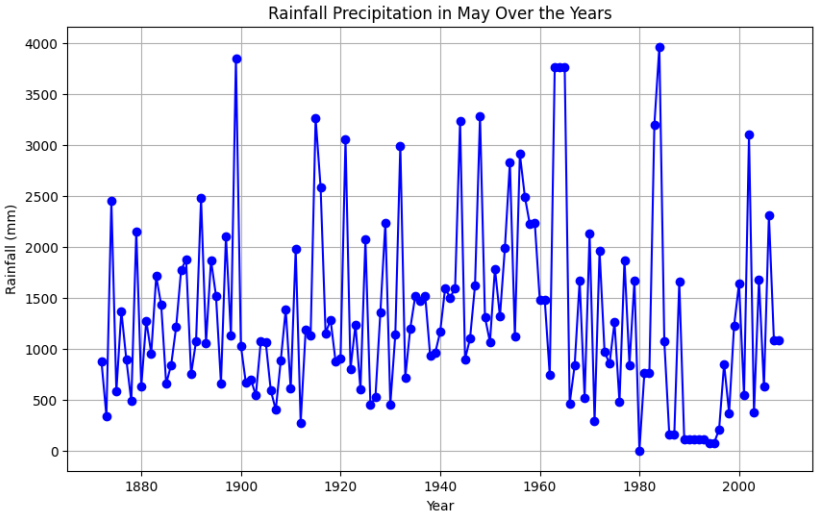

Month vs Precipitation across all years
Monthly average Precipitation(mm)
Month
0
2000
4000
6000
8000
1
2
3
4
5
6
7
8
9
10
11
12

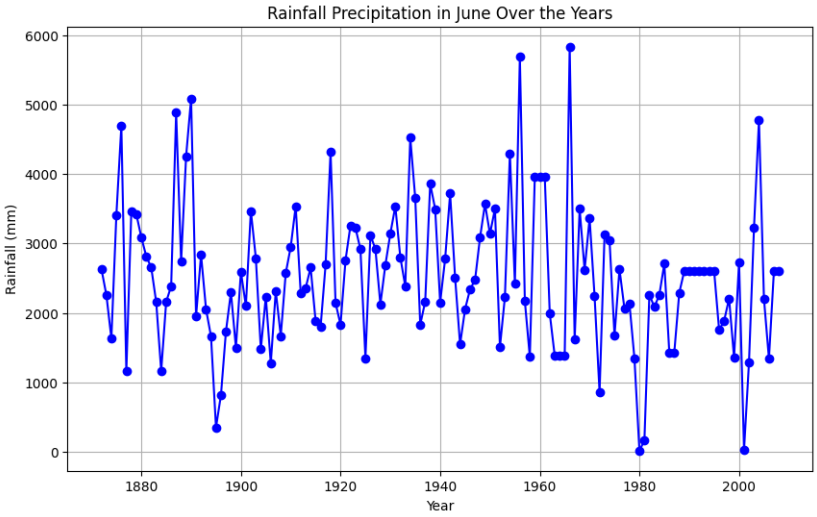

Rainfall Precipitation in January Over the Years
Rainfall (mm)
Year
0
500
1000
1500
2000
2500
3000
3500
1880
1900
1920
1940
1960
1980
2000

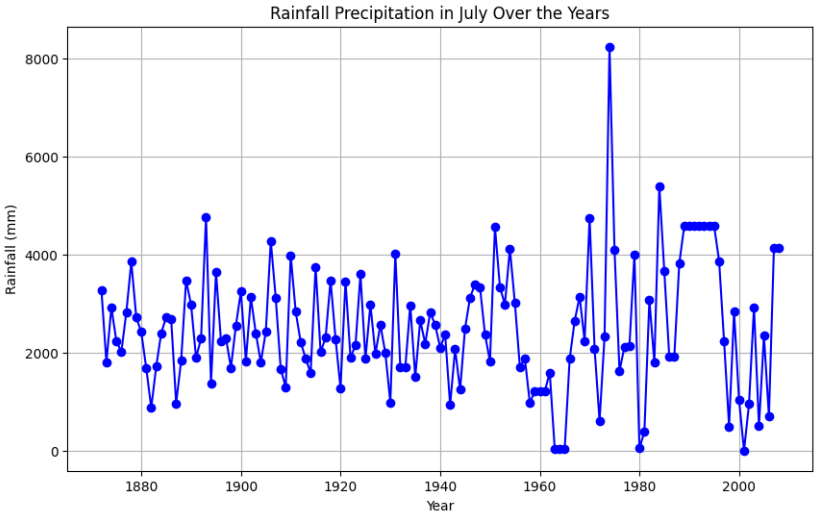

Rainfall Precipitation in February Over the Years
Rainfall (mm)
Year
0
250
500
750
1000
1250
1500
1750
2000
1880
1900
1920
1940
1960
1980
2000

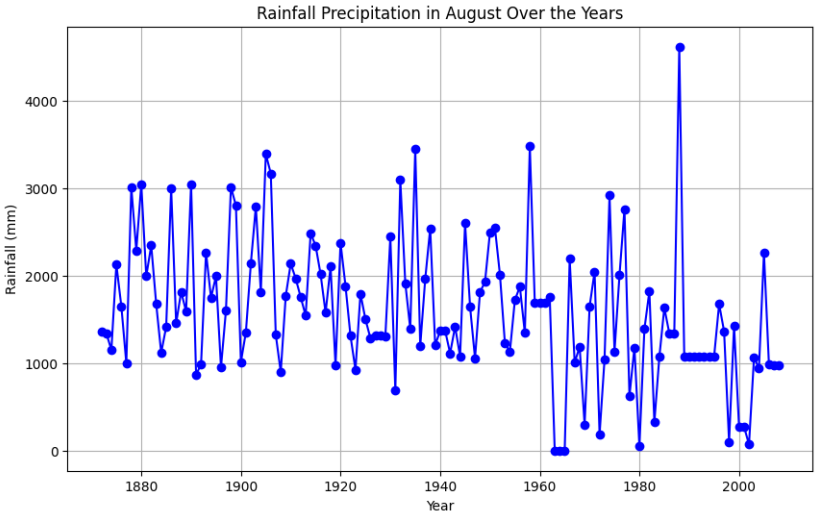

Rainfall Precipitation in March Over the Years
Rainfall (mm)
Year
0
500
1000
1500
2000
2500
3000
1880
1900
1920
1940
1960
1980
2000

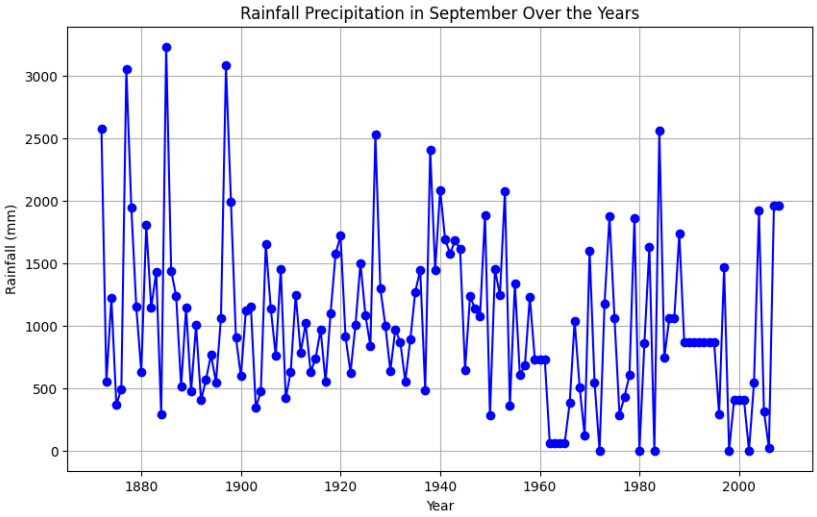

Rainfall Precipitation in April Over the Years
Rainfall (mm)
Year
0
500
1000
1500
2000
2500
3000
3500
4000
1880
1900
1920
1940
1960
1980
2000

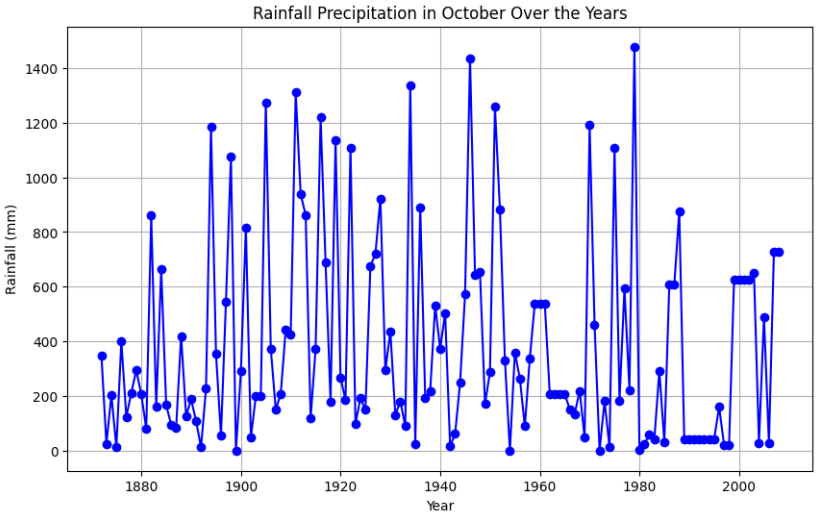

Rainfall Precipitation in May Over the Years
Rainfall (mm)
Year
0
500
1000
1500
2000
2500
3000
3500
4000
1880
1900
1920
1940
1960
1980
2000

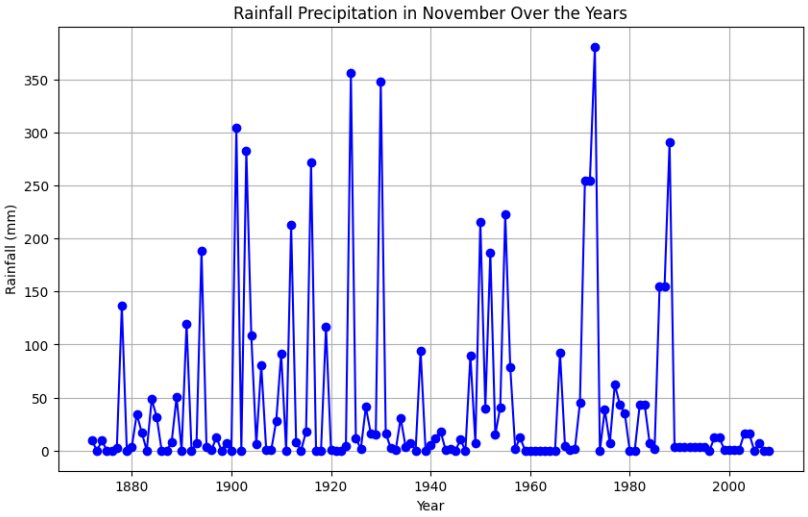

Rainfall Precipitation in June Over the Years
Rainfall (mm)
Year
0
1000
2000
3000
4000
5000
6000
1880
1900
1920
1940
1960
1980
2000

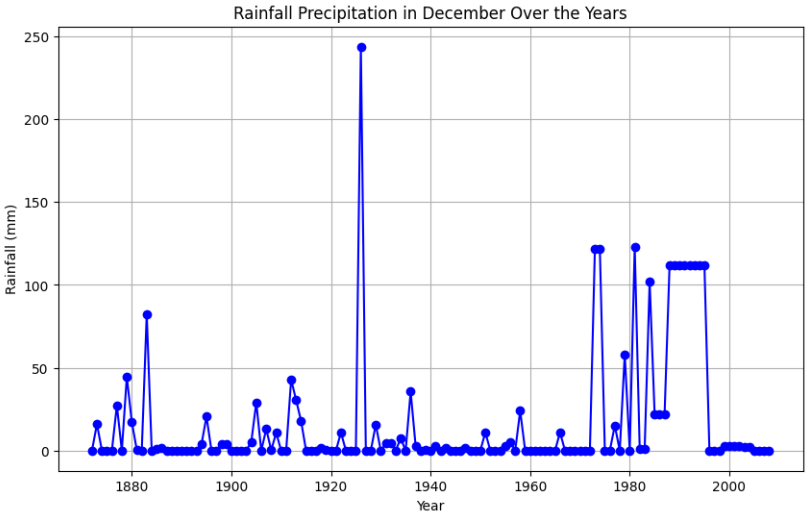

Rainfall Precipitation in July Over the Years
Rainfall (mm)
Year
0
2000
4000
6000
8000
1880
1900
1920
1940
1960
1980
2000


Rainfall Precipitation in August Over the Years
Rainfall (mm)
Year
0
1000
2000
3000
4000
1880
1900
1920
1940
1960
1980
2000


Rainfall Precipitation in September Over the Years
Rainfall (mm)
Year
0
500
1000
1500
2000
2500
3000
1880
1900
1920
1940
1960
1980
2000


Rainfall Precipitation in October Over the Years
Rainfall (mm)
Year
0
200
400
600
800
1000
1200
1400
1880
1900
1920
1940
1960
1980
2000


Rainfall Precipitation in November Over the Years
Rainfall (mm)
Year
0
50
100
150
200
250
300
350
1880
1900
1920
1940
1960
1980
2000


Rainfall Precipitation in December Over the Years
Rainfall (mm)
Year
0
50
100
150
200
250
1880
1900
1920
1940
1960
1980
2000

Chapter 4

Implementation

We are working on Cherrapunjee rainfall data collected from 1872 to 2008, The values represent monthly average rainfall

	Year	Jan	Feb	March	April	May	June	July	Aug	Sep	Oct	Nov	Dec
0	1872	13.0	8.9	294.4	735.3	879.1	2632.7	3277.9	1359.9	2577.8	348.5	9.4	0.0
1	1873	5.3	105.2	290.8	455.7	339.3	2256.0	1804.4	1338.8	551.9	24.6	0.0	16.0
2	1874	27.2	118.4	278.6	719.6	2449.3	1632.5	2923.5	1152.1	1218.2	204.0	9.4	0.0
3	1875	70.4	20.1	176.3	1380.5	578.6	3407.7	2237.5	2128.5	365.8	12.7	0.0	0.0
4	1876	0.0	41.4	446.3	810.5	1368.0	4693.9	2016.0	1651.5	493.5	400.3	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
132	2004	4.0	13.0	1234.0	1365.0	1678.0	4775.0	498.0	947.0	1917.0	28.0	NaN	NaN
133	2005	6.0	27.0	524.0	560.0	625.0	2205.0	2343.0	2262.0	314.0	490.0	0.0	0.0
134	2006	0.0	75.0	908.0	2120.0	2310.0	1341.0	699.0	986.0	21.0	27.0	7.0	NaN
135	2007	1.0	131.0	22.0	800.0	1081.0	2601.0	4133.0	974.0	1958.0	728.0	0.0	NaN
136	2008	48.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

137 rows × 13 columns

4.1 Preparation of Data

Before building the predicton models we preprocessed the data to make suitable to feed into the models. The preprocessing includes handling missing values, changing the dataset structure, splitting data into training and testing parts.

Handling Missing Values

It can be observed from the above table that some cells contains NaN values which represent missing values, one way of handling handling missing values could be removing the row from the dataset that contains missing values, but we did not want to lose any trend in the dataset so we followed a different approach, we filled the missing values with the average of that month over the years.

Training and Testing

For the experiment we divided the dataset into two parts Training and Testing part. we will be training the models using training data and we will be making prediction for the testing period and the accuracy will be measured how close the prediction is to the Testing data. For the experiment we trained models on the data from 1872 to 2007 after training the dataset for the time period we predict for the year 2008 (Testing part) and compare for results.

4.2 Building Model and Forecasting

4.2.1 Time Series Models

The two important plots while considering time series models are PACF and ACF Plots. From these plots we will be deciding the lags considered for Auto Regressive and Moving Average terms.

PACF and ACF plots for **large training data**:

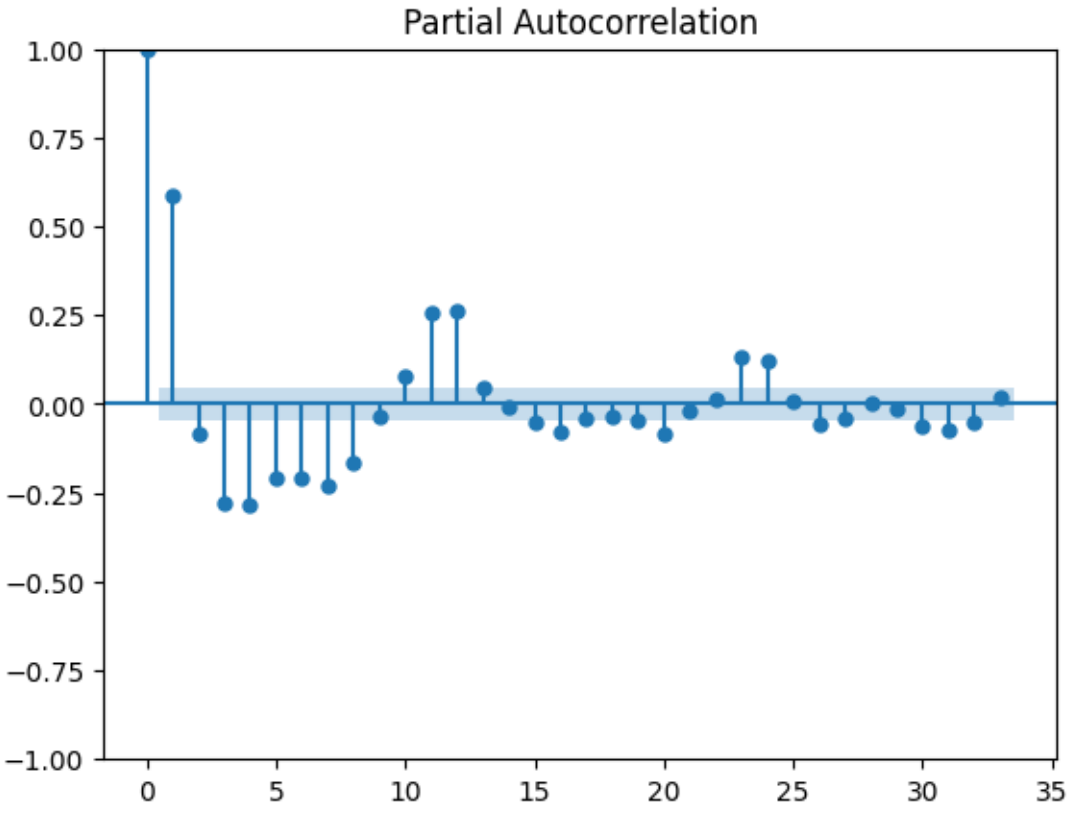


Figure 4.1: PACF plot

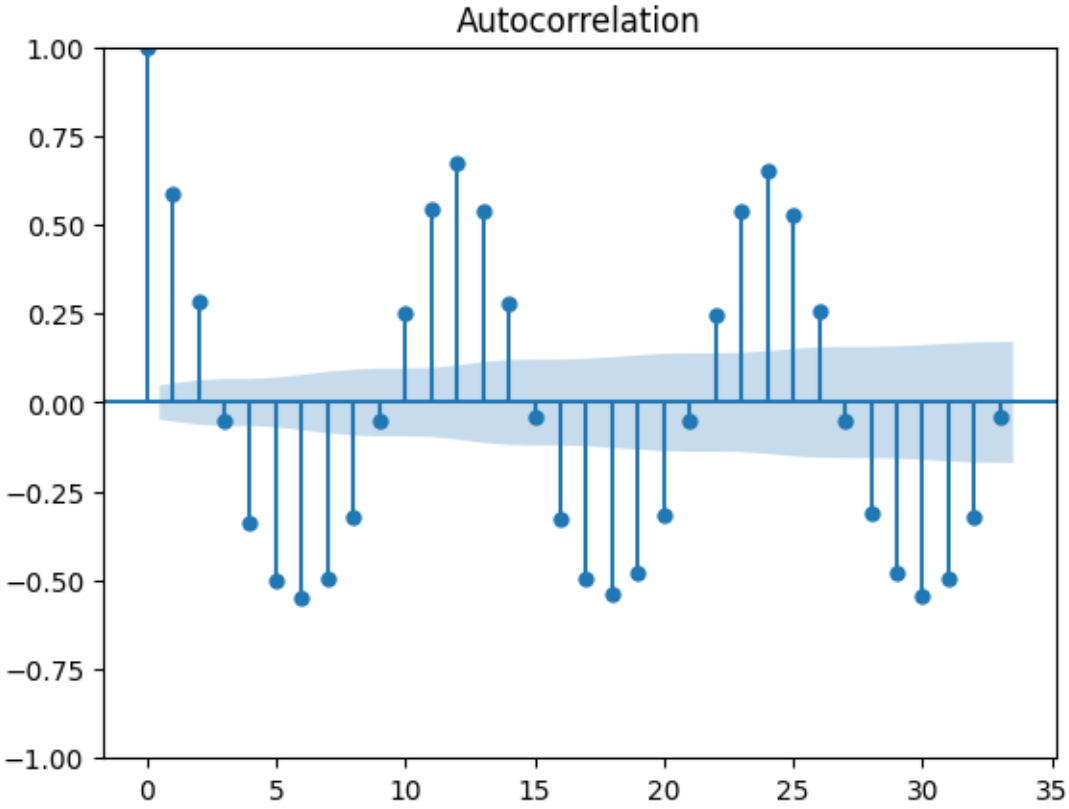


Figure 4.2: ACF plot

Auto Regressive (AR) Model

We imported the AuroRegressive Model from statsmodel (A python library) and fit the model with order 14.

The result for AutoRegressive Models is:

- Root Mean Square Error (RMSE): 221.87375122170548
- Mean Absolute Error (MAE): 183.99700326915263
- Mean Absolute Percentage Error (MAPE): 1.3047144488824554

the following figure shows the the plot of predicted and actual values

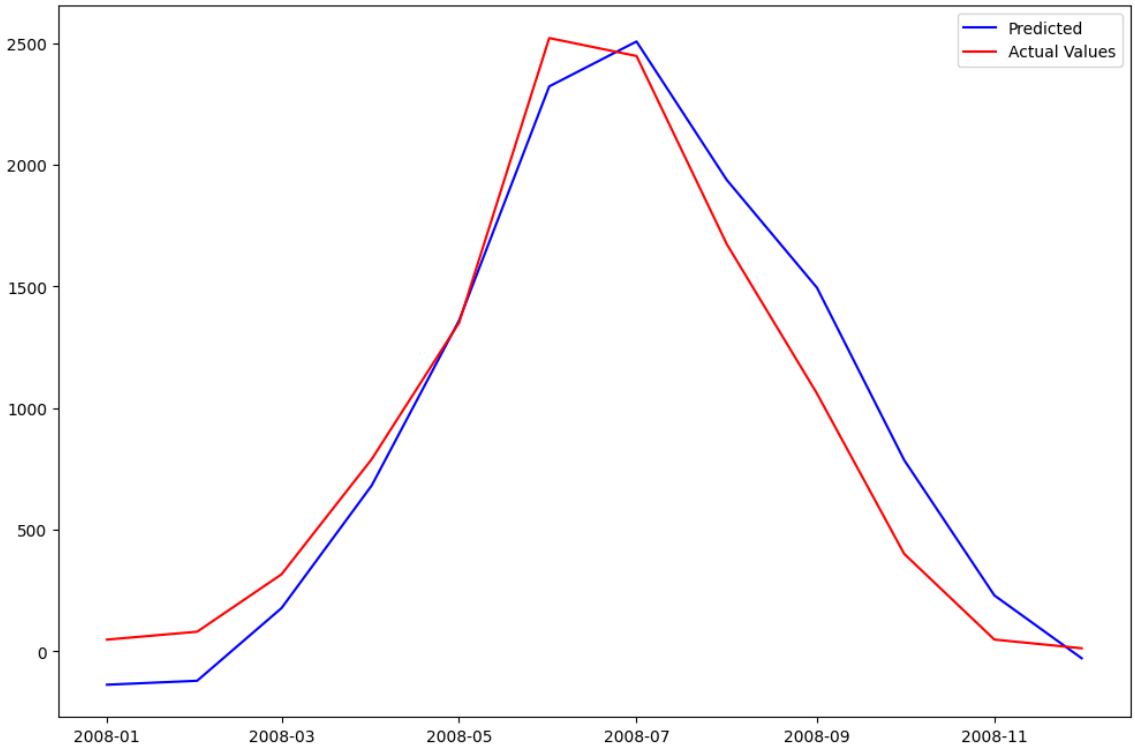


Figure 4.3: Forecasting using AR

ARIMA and Seasonal SARIMA

To train data with ARIMA the time series data has to be stationary, we tested the data with Augmented Ducky Fuller test and found out the following result

```
ADF Test Statistic : -6.874636183163738
p-value : 1.4853523964492842e-09
#Lags Used : 19
Number of Observations Used : 688
strong evidence against the null hypothesis(Ho), reject the null hypothesis. Data has no unit root and is stationary
```

Figure 4.4: Augmented Ducky Fuller Test Result

We can see that the data is stationary and there is no need for differencing to make the data stationary that makes the parameter d=0 for ARIMA Models

To find the best combination of (p,d(0),q) for ARIMA model we found

that (4,0,0) gives best AIC Score so we went ahead with it, taking seasonality S=12 for SARIMA

The result for ARIMA(4,0,0) is:

- Root Mean Square Error (RMSE): 691.1384841761496
- Mean Absolute Error (MAE): 562.3839325439789
- Mean Absolute Percentage Error (MAPE): 7.103208386478733

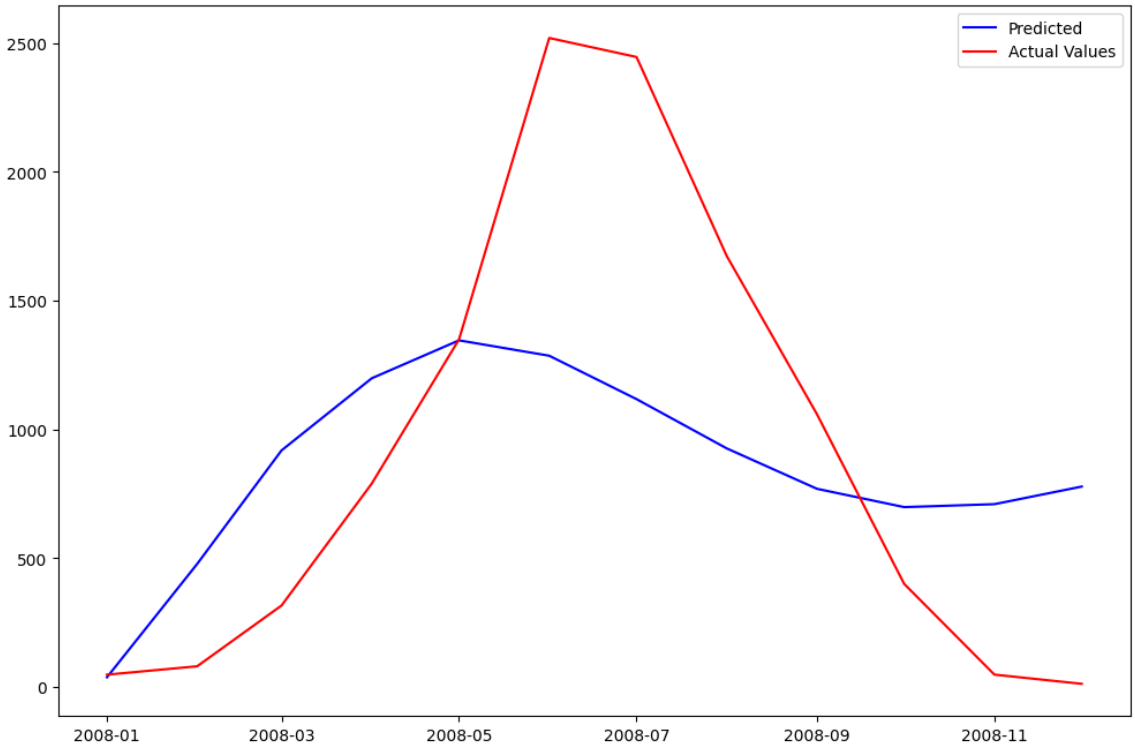


Figure 4.5: Forecasting using ARIMA

The result for SARIMAX(4,0,0)(4,0,0,12) is:

- Root Mean Square Error (RMSE):281.35196964148616
- Mean Absolute Error (MAE): 193.25819151190083
- Mean Absolute Percentage Error (MAPE): 0.5051315271391776

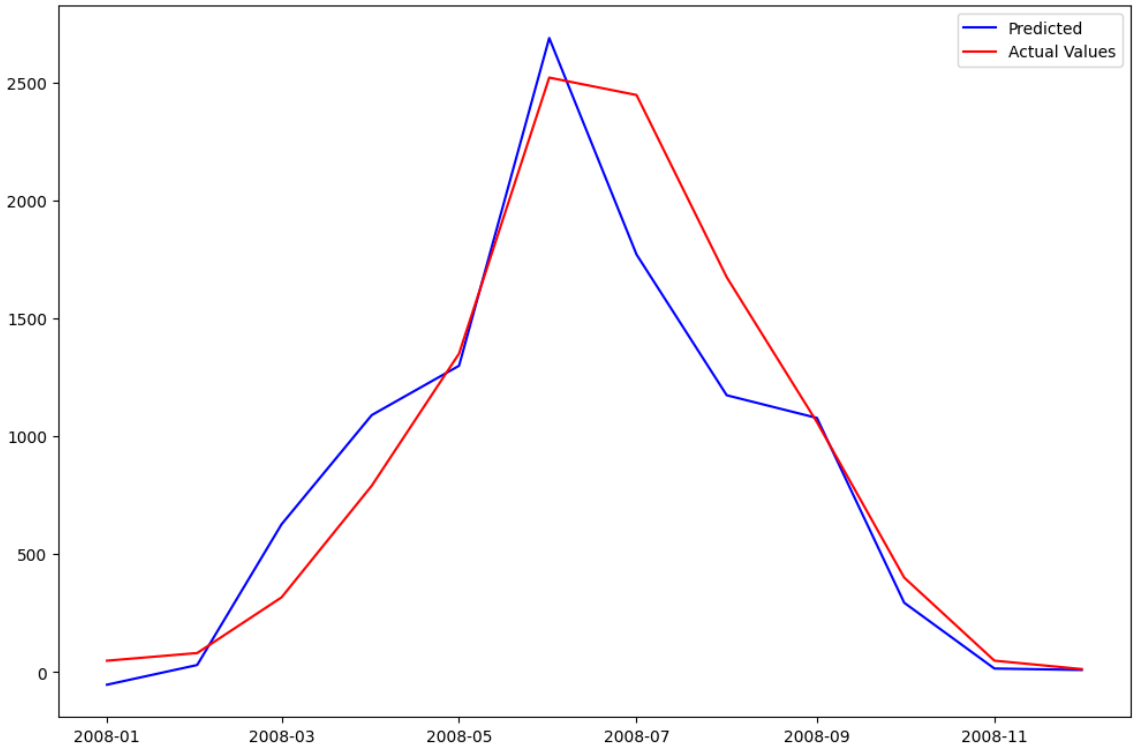


Figure 4.6: Forecasting using SARIMA

4.2.2 Regression Models

To use Regression Models we created some 6 features for a certain point using the data available before it. The first 6 rows do not have some values because we used the values to construct features and to make the regression models we discarded the first 6 rows.

To fit the dataset into regression models Six months of past precipitation data were extracted from dataset[4.7] and reshaped into column vectors. These monthly precipitation arrays were then concatenated horizontally to form a feature matrix (finalX), where each row represents the precipitation values for the past six months. This feature matrix is prepared for modeling or further analysis.

The dataset was divided into training and test sets, with the last 30

Month	Precipitation	Precipitation_Last_Month	Precipitation_Last2Month	Precipitation_Last3Month	Precipitation_Last4Month	Precipitation_Last5Month	Precipitation_Last6Month
1872-01-01	13.000000	NaN	NaN	NaN	NaN	NaN	NaN
1872-02-01	8.900000	13.000000	NaN	NaN	NaN	NaN	NaN
1872-03-01	294.400000	8.900000	13.000000	NaN	NaN	NaN	NaN
1872-04-01	735.300000	294.400000	8.900000	13.000000	NaN	NaN	NaN
1872-05-01	879.100000	735.300000	294.400000	8.900000	13.000000	NaN	NaN
...	...	...	...	...	...	...	...
2008-08-01	1673.929032	2446.588000	2520.403200	1350.040945	789.878462	316.542308	80.493182
2008-09-01	1059.699180	1673.929032	2446.588000	2520.403200	1350.040945	789.878462	316.542308
2008-10-01	400.368333	1059.699180	1673.929032	2446.588000	2520.403200	1350.040945	789.878462
2008-11-01	48.150427	400.368333	1059.699180	1673.929032	2446.588000	2520.403200	1350.040945
2008-12-01	12.470476	48.150427	400.368333	1059.699180	1673.929032	2446.588000	2520.403200

1644 rows × 7 columns

Figure 4.7: Dataset After Feature Engineering

observations used as the test set to evaluate model performance. The feature matrix (finalX) was split into Xtrain (all rows except the last 30) and Xtest (last 30 rows). Similarly, the target variable (y) was split into ytrain and ytest to align with the training and testing features

The result for Linear Regression is:

- Root Mean Square Error (RMSE): 374.7968655796515
- R2 Score: 0.7418046834303538

The result for Random Forest Regression is:

- Root Mean Square Error (RMSE): 136.5943107749893
- R2 Score: 0.9770301388853283

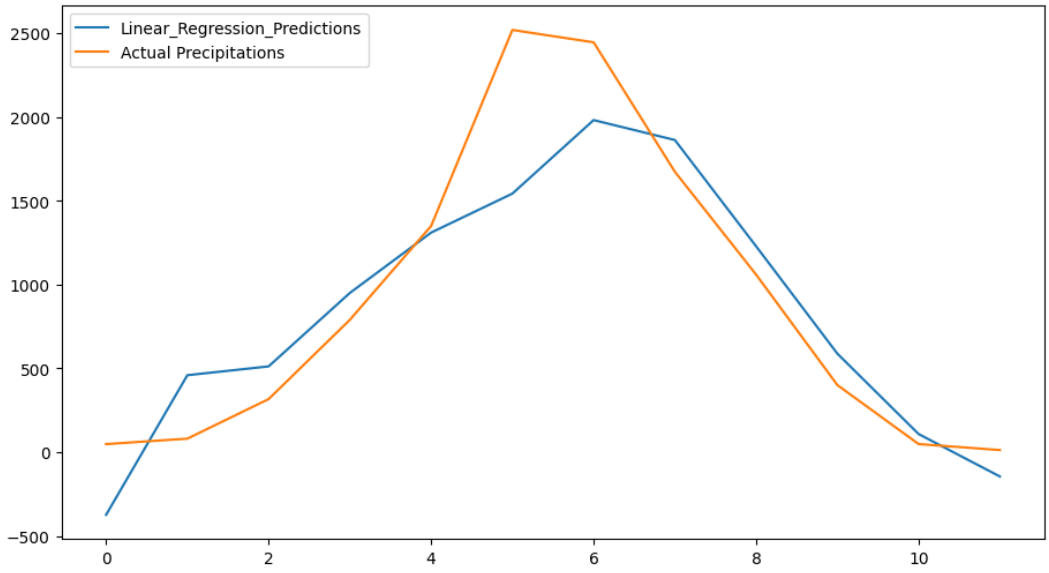


Figure 4.8: Forecasting using Linear Regression

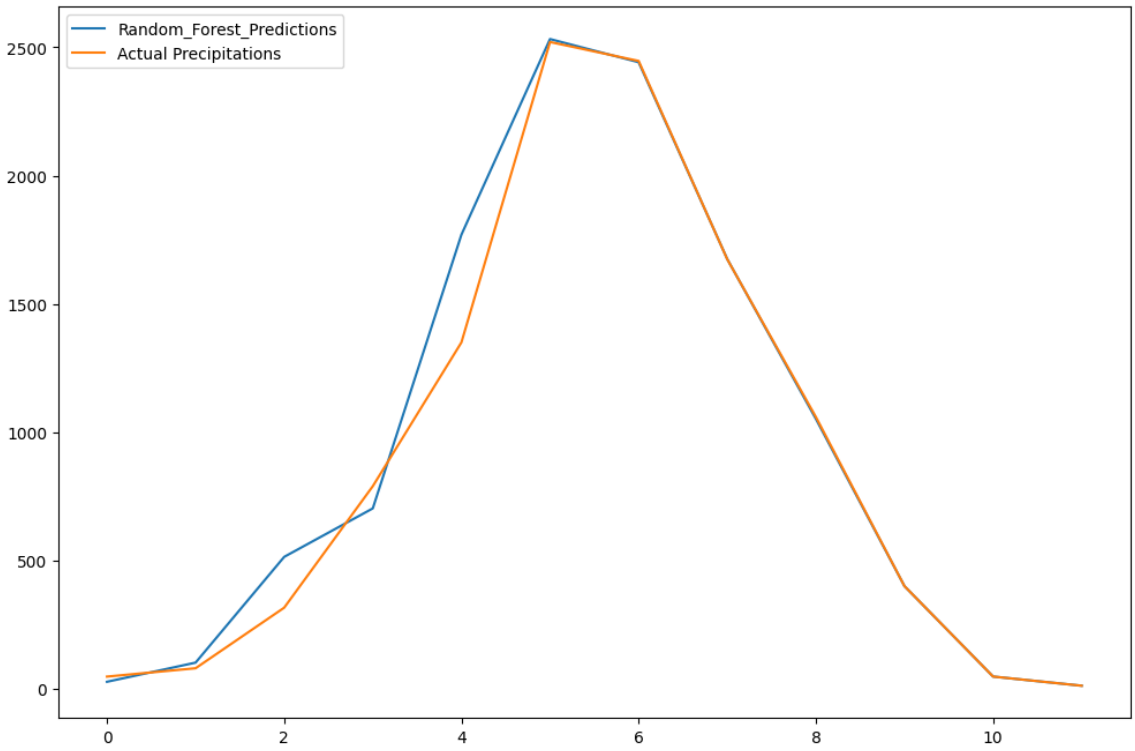


Figure 4.9: Forecasting using Random Forest Regression

4.2.3 LSTM

The dataset is first split into training and test sets, with the training data scaled to a 0-1 range using MinMaxScaler(from sklearn python libary). An LSTM (using Tensorflow library) model is then built to capture temporal dependencies in the rainfall data, where it takes in sequences of past observations and learns patterns to predict future values. The model is trained on these sequences for 50 epochs, optimizing to reduce mean squared error (MSE).

To make predictions, the trained model first generates a prediction from the final 12 values in the training data. This serves as a starting point for forecasting into the test period. For each subsequent step, the model uses a 'sliding window' approach, where the input to the model is dynamically updated by appending the latest prediction and removing the oldest data point from the sequence. This iterative process continues for the length of the test set, creating a series of forecasted values. Each prediction is then stored in which can be compared against the actual values for evaluation.

The result for LSTM is:

- Root Mean Square Error (RMSE): 131.75902813597824
- Mean Absolute Error (MAE): 108.75964872869262
- Mean Absolute Percentage Error (MAE): 1.520719767115741

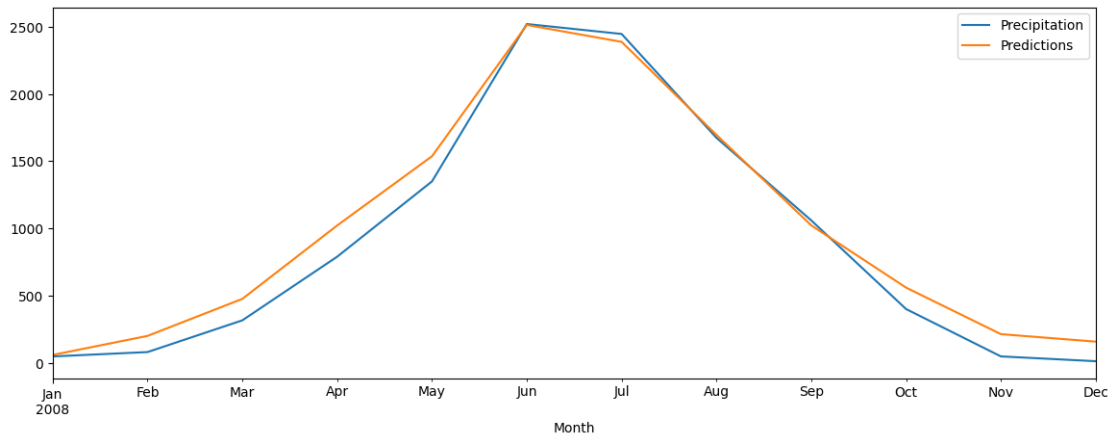


Figure 4.10: Forecasting using LSTM

4.3 Time Series Models Considering Smaller Data

Previously we trained the data on rainfall data from 1872 to 2007 and made the prediction for 2008, in this section we trained the time series models from 1950 to 1007 and make predictions for 2008

Fig 4.11 shows PACF and ACF plots for **large training data**:

4.3.1 Auto Regressive (AR) Model

similarly as large data we look at the PACF Plot to consider number of lags. Here we consider AR of 14 as well

The result for AutoRegressive Model on less data is:

- Root Mean Square Error (RMSE): 244.21618878104675
- Mean Absolute Error (MAE): 196.960509611807
- Mean Absolute Percentage Error (MAPE): 1.248962328556654

Fig 4.12 shows the plot of predicted and actual values using AR Model on less data.

4.3.2 ARIMA and SARIMA

To use ARIMA model on data the data has to be stationary so we again did ADF test and found out that the data is stationary again and that makes the differencing d=0, We again tried different combinations of (p,d(0),q) values for arima and found that (2,0,3) gives the best AIC Score so we went ahead with it.

The result for ARIMA(2,0,3) is:

- Root Mean Square Error (RMSE): 336.44754150451826
- Mean Absolute Error (MAE): 268.45458765124073
- Mean Absolute Percentage Error (MAPE): 2.0028316691643298

The result for SARIMAX(2,0,3)(2,0,3,12) is:

- Root Mean Square Error (RMSE):168.95827039556045
- Mean Absolute Error (MAE): 116.53287759325258
- Mean Absolute Percentage Error (MAPE): 0.401581538287813

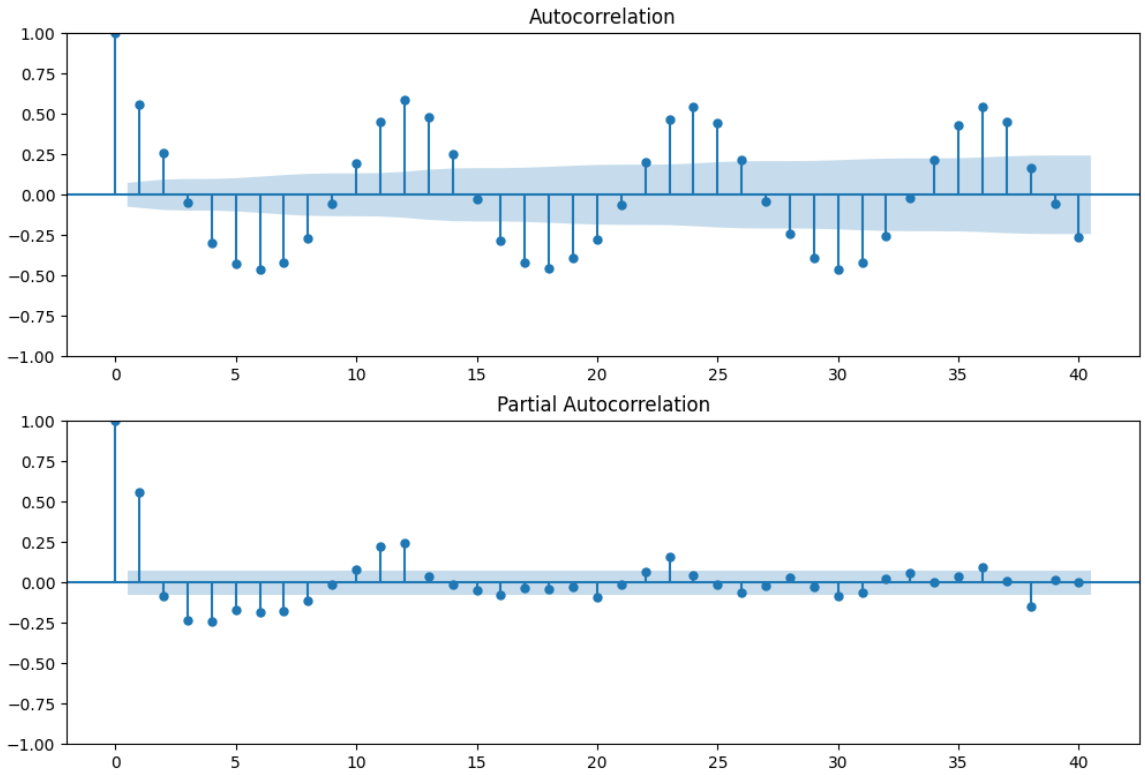


Figure 4.11: Forecasting using AR on less data

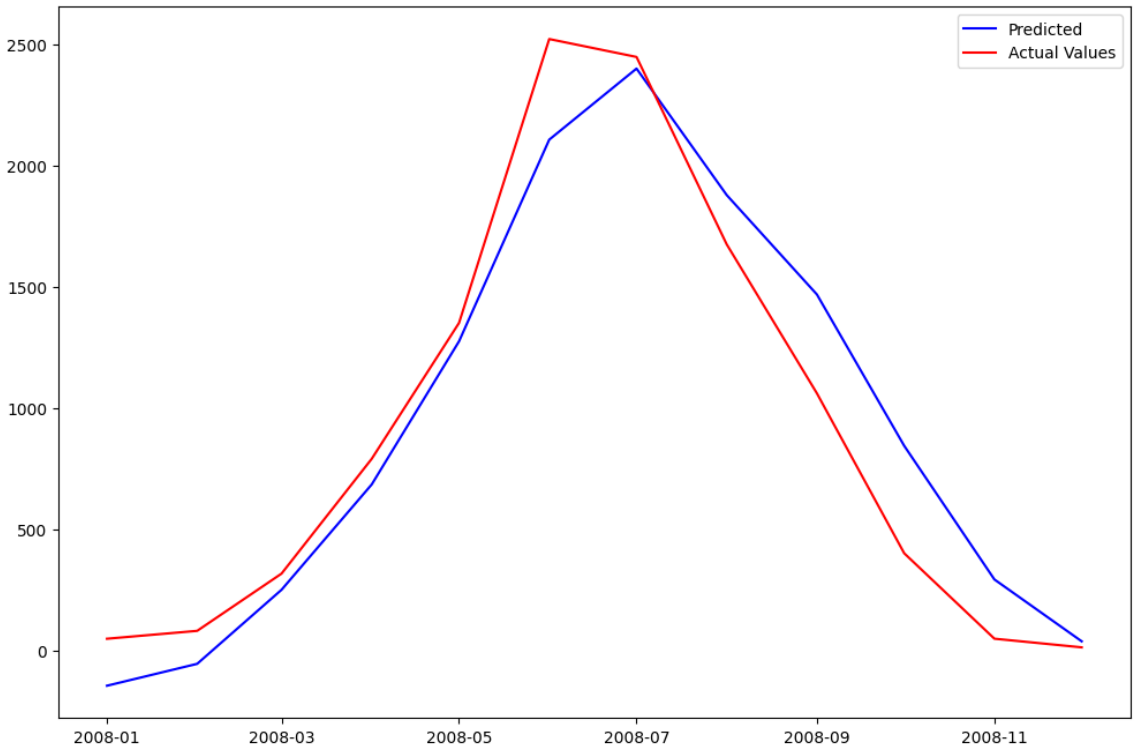


Figure 4.12: Forecasting using AR on less data

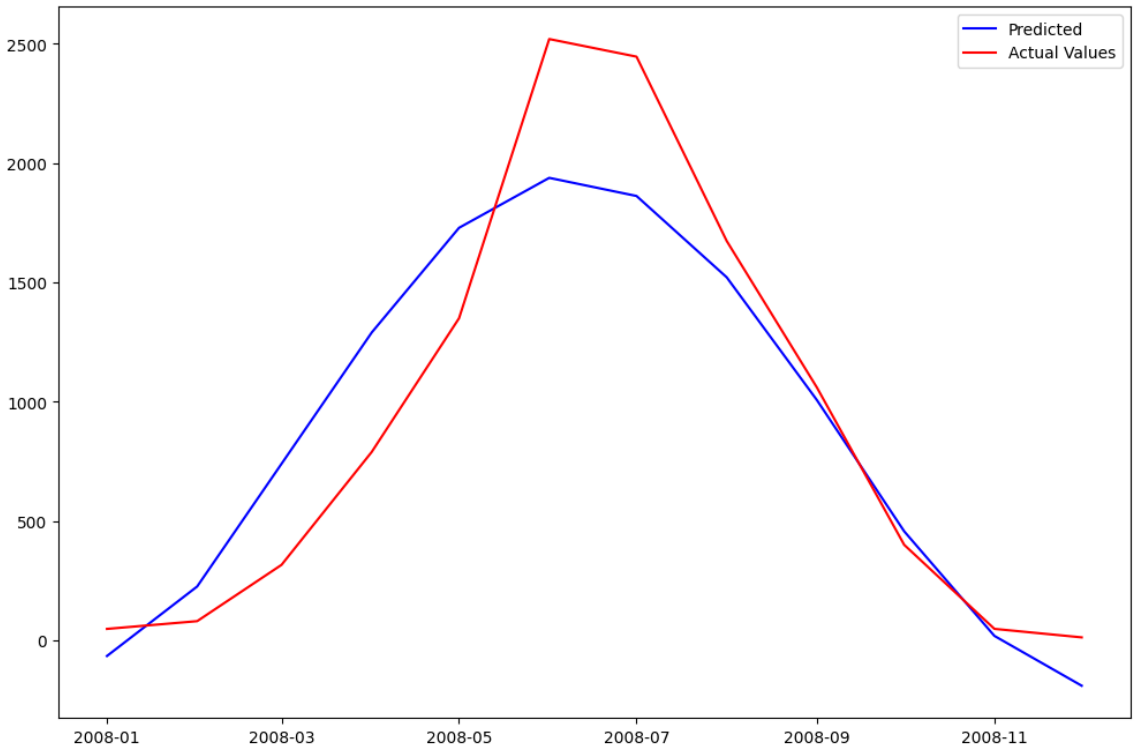


Figure 4.13: Forecasting using ARIMA on less data

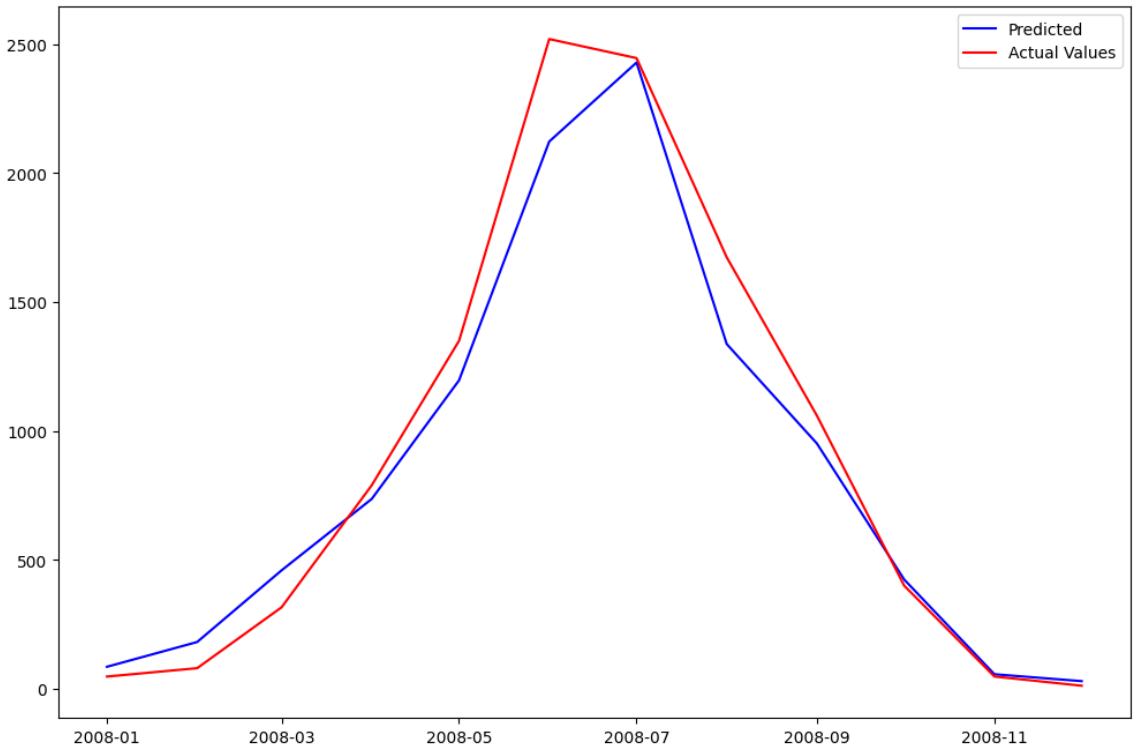


Figure 4.14: Forecasting using SARIMA on Less Data

Chapter 5

Conclusion

The project has provided us various insights of models and approaches used in rainfall prediction using time series data. In this section we draw conclusion based on the following metric Root Mean Square Error(RMSE), Mean Absolute Error(MAE), Mean Absolute Percentage Error(MAPE) for Time Series Models and LSTM, and the metrics used to evaluate Regression Models are RMSE and R2 Score.

5.1 Model Performance Analysis

We made the prediction for two cases by considering two types of dataset, the below analysis is for both the cases

5.1.1 Larger Dataset (from 1872 to 2008)

- **Time Series Models and LSTM**
  - Based on RMSE LSTM performed best among all the models and AR models performed better than both ARIMA and SARIMA

model, which is less expected. ARIMA performed worst of all.
  - Base on MAPE SARIMA performed best among all with only 0.5 percent and ARIMA performed worst with around 7 percent error.
  - Based on MAE LSTM performed best among all and ARIMA performed worst among all.

- **Regression Models**

  - Here both based on RMSE and R2 Score Linear Regression is performing worst with only around 0.74 R2 Score while Random Forest Regression's R2 Score is around 0.97. Result using Random Forest model is almost accurate for half of the year as observed from fig 4.9.

5.1.2 Smaller Dataset (from 1952 to 2008) on Time Series Models

- In This case using all metrics RMSE, MAE, MAPE SARIMA performs best among all with MAPE of only 0.4 percent.
- ARIMA still performs worst of all the models but it's performance increased a lot from previous case when we considered large dataset it can be observed from MAPE being shifted from 7 percent to 2 percent and it improved in other metrics too.
- AR model improved it's MAPE but it's RMSE and MAE increased.

5.2 General Conclusion

It can be observed from the Model Performance Analysis that using regression models for predicting time series data could be beneficial and if it is with good feature engineering.

LSTM performed best and the reason could be it works on Recurrent Neural Network.

Theoretically ARIMA should perform better than AR Model but it performs worse than AR Model, reason could be on large dataset ARIMA Model fails to capture the trend correctly and tend to over generalize. But the performance of ARIMA increased in smaller dataset the reason could be the data has less changing trends in small time period.

SARIMA performed better than other Time Series Models the reason being it considers an extra seasonal factor for observing the trends.

These findings gives us a better understanding in rainfall prediction which can be used in future model selection for rainfall prediction of Cherrapunjee or other region.

Chapter 6

Future Work

Introductory lines...