Time Series Analysis of Earth Surface Temperature (Delhi)

Zeel Ghori
202201287@daiict.ac.in

Naisargi Patel
202201291@daiict.ac.in

Akshat Jindal
202201299@daiict.ac.in

***Abstract*—This report presents a simple and clear analysis of temperature data for Delhi using time series methods. The study focuses on finding unusual temperature values (outliers), checking the shape of data (kurtosis and skewness), observing warming trends, removing seasonal effects using different methods, and using the SARIMA model for forecasting. The data was analyzed using Jupyter Notebooks and showed that Delhi is getting warmer over time. Several techniques such as moving average, seasonal decomposition, and differencing were used to better understand the data. The SARIMA model was then used to predict future temperatures. Deep Learning models were implemented to improve the prediction.**

***Index Terms*—Time Series Analysis, Outliers, Skewness, Warming Trend, Deseasonalization, SARIMA Model, Deep Learning, LSTM, RNN, GRU**

I. Introduction

Understanding temperature changes is important to monitor the climate and plan for the future. This report explains how we studied monthly temperature data of Delhi [1] using time series analysis. We used simple tools and methods to look at how temperatures have changed and to predict how they might change in the future.

II. Data Interpretation

We have specifically used data for Delhi, India, only to analyze the surface temperature of a specific region. The range of average monthly temperature data is dated from January 1950 to September 2013. The data set used in this project is taken from [1].

A. Data format

The dataset used in this study, titled GlobalLandTemperaturesByMajorCity.csv, contains long-term monthly average land surface temperatures for major global cities, including Delhi. Provided in CSVformat, the dataset comprises over 23,000 records and 7 columns. The key columns include the observation date (dt), average temperature (AverageTemperature), uncertainty (AverageTemperatureUncertainty), city, country, and geographical coordinates (Latitude and Longitude). The average temperature is recorded in degrees Celsius.

B. Data Resolution and Granularity

Temperature values are recorded at a monthly resolution, allowing decadal and seasonal aggregation for trend analysis.

C. Data cleaning and processing

Rows with missing values or NaN values in AverageMonthlyTemperature column were estimated using interpolation technique.

D. Temporal coverage

Although the dataset spans from 1743 to 2013, this study focuses on the period from 1950 to 2013 due to improved data quality and consistency. Data prior to 1950 contains more missing values and higher uncertainty, reflecting limited observational coverage. Starting from 1950 ensures a reliable baseline and strengthens the validity of long-term trend analysis.

III. Understanding time series using plots

A. Month wise trend of average temperature

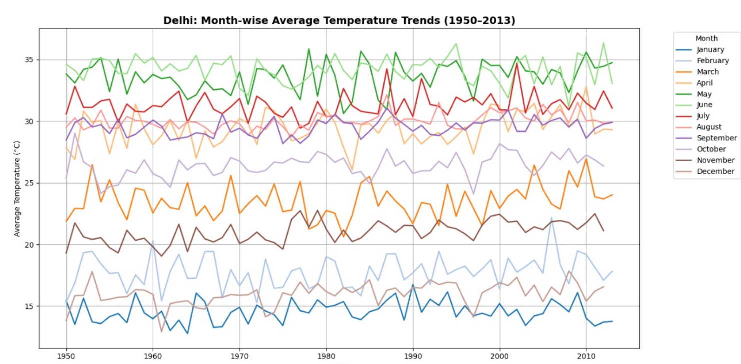

Fig. 1: This line chart illustrates the average temperature trends for each month across multiple years, showing seasonal variations and long-term patterns in temperature.

B. Variation of temperature in particular years

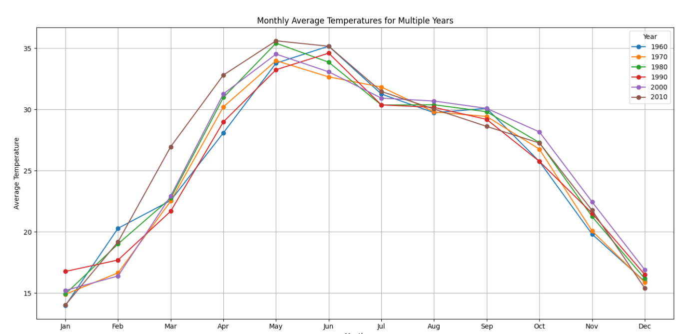

Fig. 2: This line chart displays the monthly average temperatures across six selected years—1960, 1970, 1980, 1990, 2000, and 2010—highlighting seasonal changes and year-to-year variations.

C. Distribution characteristics: Skewness and Kurtosis

Skewness was analyzed yearly and monthly to assess asymmetry in temperature distributions—positive skew showed more hot extremes, while negative skew pointed to cold events. Kurtosis measured the "tailedness" or outliers—low values indicated flatter, stable temperatures, whereas high kurtosis signaled frequent extreme temperature events.

1) Skewness: The skewness plot shows consistently negative values across all years, indicating leftskewed temperature distributions with occasional colder-than-average days and more warmer days than average. This trend persists despite rising average temperatures and becomes more pronounced post-2000.

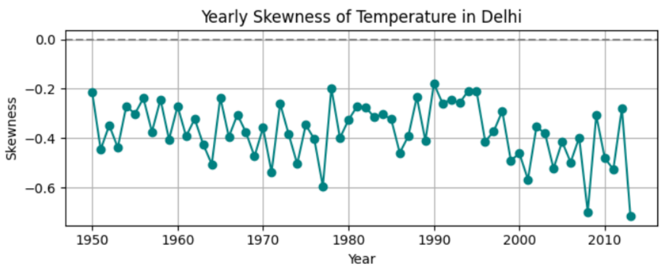

Fig. 3: Yearly skewness of temperature in Delhi from 1950 to 2013.

2) Kurtosis: The kurtosis plot shows mostly platykurtic distributions (values < 3), indicating flatter temperature profiles with few extremes. Values near -1 suggest limited outliers, though a slight recent rise may signal growing variability due to more frequent heatwaves and cold spells.

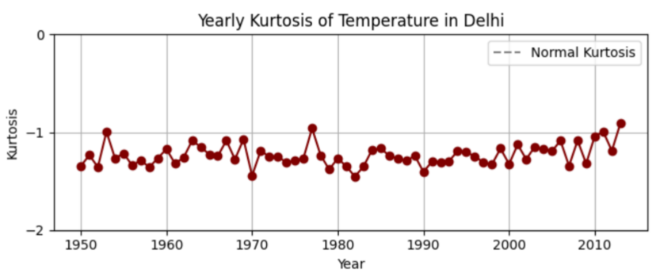

Fig. 4: Yearly kurtosis of temperature in Delhi from 1950 to 2013.

D. Long term warming trends

A non-linear regression analysis was performed on Delhi's annual average temperature data (1950–2013) to identify long-term warming trends and visualize decadal variations using statistical modeling.

- Each decade is color-coded in the plot, with solid lines representing yearly averages and a dashed black line indicating the non-linear warming trend.
- **1950s–1970s:** Temperatures remain stable with a nearly linear trend.
- **1980s–1990s:** Increased variability appears, and the trend begins to curve upward.
- **2000s–2010s:** A clear rise in temperature is observed, with the trend accelerating.
- **Overall:** Initially, the regression line appears almost linear, but as time progresses, it curves upward steeply, highlighting a significant and accelerating warming trend in Delhi from 1950 to 2013.

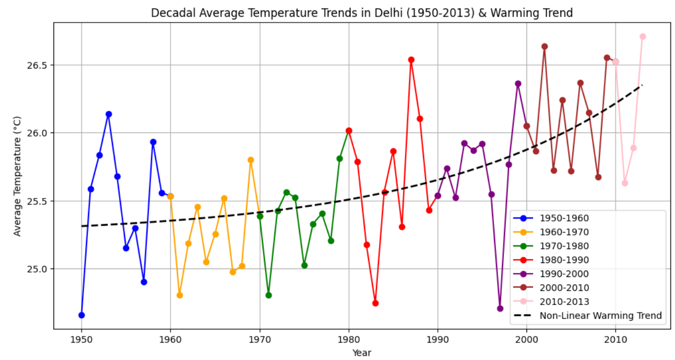

Fig. 5: Average yearly temperatures for Delhi (1950-2013) grouped by decade, showing a clear warming trend with a non-linear fit.

A 4^{th}degree polynomial regression was applied per decade to Delhi's annual average temperature data (1950–2013) to reveal non-linear trends within each decade. Each decade is color-coded in the plot, showing actual data as solid lines and trends as dashed lines, helping highlight subtler patterns missed by a single long-term trend.

- **1950s–1970s:** Relatively stable with minor variations; trends are mostly flat.
- **1980s–1990s:** Increased variability with noticeable dips and spikes; early signs of warming appear.
- **2000s–2010s:** A clear upward trend becomes evident, indicating stronger and more consistent warming.

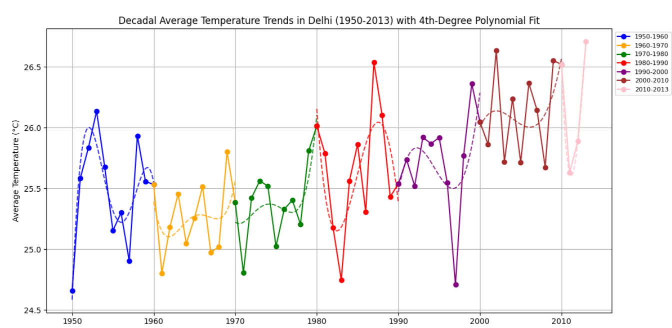

Fig. 6: Decadal average temperature trends in Delhi (1950–2013) with a 4th-degree polynomial fit showing a steadily rising and accelerating warming pattern.

E. Overview

The analysis of monthly average temperatures across multiple years reveals both consistent seasonal patterns and evolving long-term trends. Two line plots were analyzed—one showing annual temperature profiles from 1950 to 2012, and another comparing selected years from 1960 to 2010. These visualizations offer meaningful insights into how temperatures have changed over time.

F. Key Insights

1) **Seasonal Temperature Cycle is Consistent:**
   - Across all years, a clear seasonal pattern is observed.
   - Temperatures rise from January, peak around May–June, and decline toward December.
   - January and December are the coolest months, while May and June are the hottest.
2) **Long-Term Warming Trend:**
   - A gradual increase in average temperatures is visible across decades.
   - Summer months (April to June) especially show a noticeable rise in peak temperatures in recent decades.
   - Winter months also exhibit a warming trend, albeit to a lesser degree.
3) **Decadal Comparison Reveals Rising Extremes:**
   - The second chart (1960–2010) demonstrates that recent years, especially 2010, experienced higher average temperatures throughout the year.
   - Earlier decades such as 1960 and 1970 reflect cooler overall temperatures.
   - This suggests a rising temperature baseline over time.
4) **Greater Fluctuation in Recent Decades:**
   - Post-1980s data shows more variability from year to year, particularly in May to July.
   - This increased variability may reflect growing climate instability and changing weather patterns.
5) **Evidence of Climate Change:**
   - The rising trend in both average and extreme temperatures, coupled with seasonal changes, is consistent with indicators of climate change.
   - This has potential implications for agriculture, water resources, energy demand, and public health.

IV. Data Analysis

A. Outlier detection and Potential causes

Outlier detection was conducted across the entire monthly temperature dataset for Delhi, spanning from 1950 to 2013. The goal was to identify extreme temperature values that could distort trend analysis or affect model performance. The Interquartile Range (IQR) method was used for this purpose. Possible reasons for the presence of outliers include:

- **Absence of Western Disturbances:** Instances such as February 1960 and February 2006 exhibited unusually high temperatures due to the lack of Western Disturbances, resulting in clear skies and increased solar radiation. [2]
- **Prolonged Clear Skies and Lack of Rainfall:** Heatwaves like the one in April 2010 were intensified by extended periods of clear skies and minimal pre-monsoon rainfall, preventing natural cooling and pushing temperatures to record highs. [3]
- **Delayed Monsoon Withdrawal:** In October 1951, the highest recorded average temperature for the month was observed, largely attributed to the late withdrawal of the monsoon season and persistent clear skies. [4]
- **Delayed Winter Onset and Reduced Cold Influx:** Warmer-than-normal conditions in December 2008 are linked to the delayed onset of winter and a reduction

Month-Year		Month-Year
April 1952		February 2006, April 1983, April 2010
Feb 1961		July 1987,July 2002,August 1987
March 1978		August 1993 ,August 2009, October 1951
May 1978		October 1954 ,October 1997, December 1961
May 1984		December 1973, December 1997, December 2008
July 1988		
June 2008		

Fig. 7: Potential Outliers and Outliers: On the left are months showing sudden deviations from previous years and on the right are outlier months identified using the IQR method.

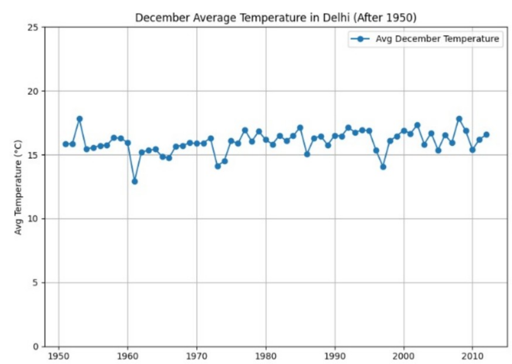

Fig. 8: December Temperature Trends in Delhi (Post-1950). Year-wise average December temperatures showing general stability with occasional dips.

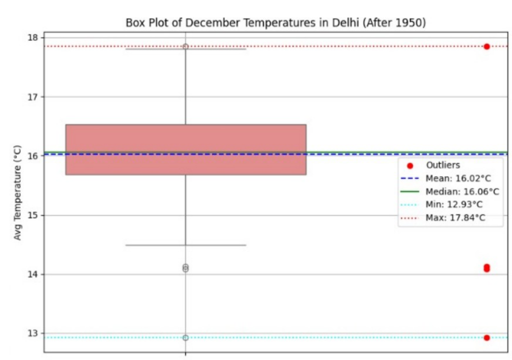

Fig. 9: December Temperature Trends in Delhi (Post-1950). Box plot showing temperature distribution, outliers, and key statistics like mean and median.

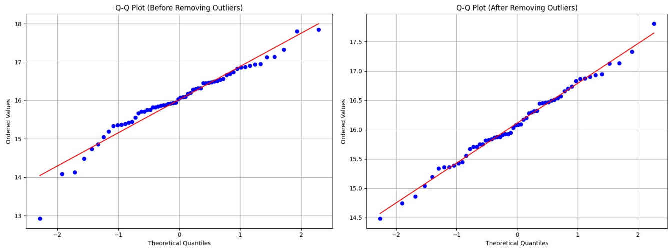

Fig. 10: Q-Q plots before and after outlier removal show improved alignment with the normal distribution after outlier removal.

To ensure the reliability of the temperature dataset, outlier detection and normalization were applied. The Interquartile Range (IQR) method was used to remove outliers for each month, with box plots confirming anomalous values.
After outlier removal, normality checks were performed using statistical measures and visual tools like Q-Q plots and histograms. If the data didn't follow a normal distribution, transformations such as logarithmic, square root, or Box-Cox were applied. These steps were essential to meet assumptions for subsequent analyses and improve model performance.

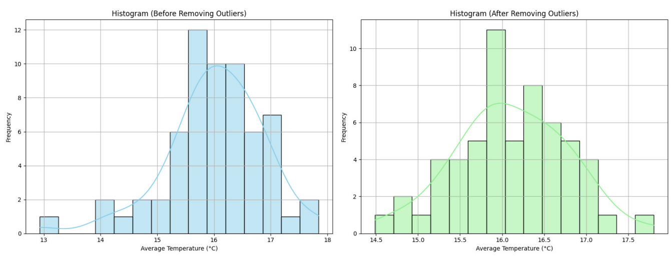

Fig. 11: Histograms before and after outlier removal show a more symmetric and bell-shaped distribution post-removal, indicating improved normality.

V. Rolling Statistics

A. *Rolling mean and Rolling standard deviation*

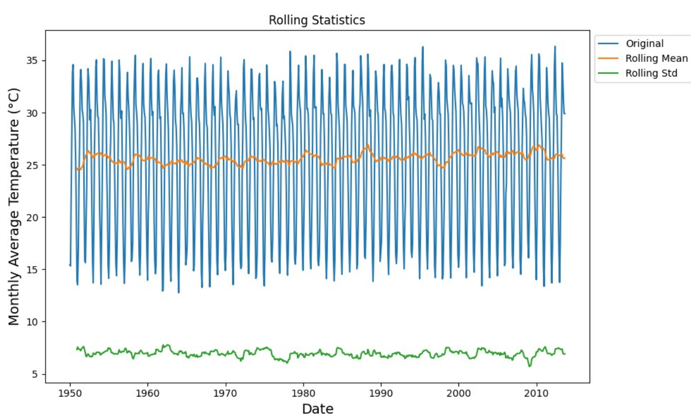

Fig. 12: Yearly average temperature in Delhi (1950–2013), with 12-month rolling mean and standard deviation shown to highlight trend and stationarity.

B. *Moving averages*

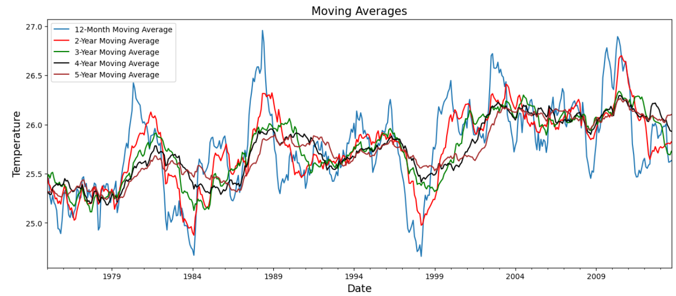

Fig. 13: Comparison of Moving Averages over Time. The plot shows various smoothing windows applied to the monthly average temperature data, including 12-month, 2-year, 3-year, 4-year, and 5-year moving averages. As the window size increases, the temperature trend becomes smoother, effectively filtering out short-term fluctuations and highlighting long-term trends and patterns in the data.

From the graph, it is evident that:

- The rolling mean remains relatively stable throughout the observed period, suggesting that there is no significant long-term upward or downward trend in average monthly temperatures.
- The rolling standard deviation also remains fairly consistent, with only minor fluctuations, indicating that the variance of the time series does not change dramatically over time.

These observations imply that the dataset may be considered **weakly stationary** in terms of mean and variance, although the pronounced seasonality component may require differencing or seasonal decomposition for further time series modeling, such as ARIMA or SARIMA.

VI. Statistics test for stationarity

A. *ADF test result*

The Augmented Dickey-Fuller (ADF) test was conducted to check the stationarity of the time series. The ADF statistic was found to be **-6.0273**, which is significantly lower than all the critical values at 1%, 5%, and 10% significance levels. Additionally, the **p-value was 1.45e-07**, which is much lower than the conventional threshold of 0.05. **Conclusion:** Since the ADF statistic is less than the critical values and the p-value is very small, we reject the null hypothesis of non-stationarity. Thus, we conclude that the time series is **stationary**.

B. *KPSS test result*

The Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test was used to verify the stationarity of the time series. The KPSS test statistic was found to be **0.3562**, which is below the critical value at the 5% significance level (**0.463**). The corresponding **p-value was 0.096**, which is greater than the commonly used threshold of 0.05.
Conclusion: Since the test statistic is lower than the 5% critical value and the p-value is greater than 0.05, we **fail**

to reject the null hypothesis of stationarity. Therefore, the KPSS test also indicates that the time series is **stationary**.

VII. Decomposition of Time Series

A. Additive Decomposition

Additive decomposition is a technique used to analyze a time series by separating it into three distinct components. Mathematically, the additive decomposition assumes the time series value at time t, denoted as Y_t, is composed of the sum of the three components:

$$Y_t = \text{Trend}_t + \text{Seasonality}_t + \text{Residual}_t$$

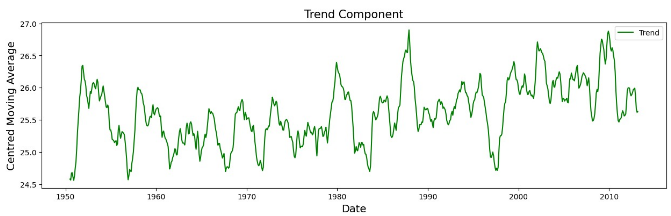

Fig. 14: Trend component showing a gradual increase in average temperature in Delhi from 1950 to 2013.

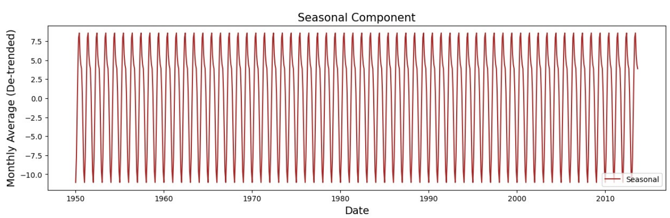

Fig. 15: Seasonal component showing consistent annual temperature patterns in Delhi from 1950 to 2013.

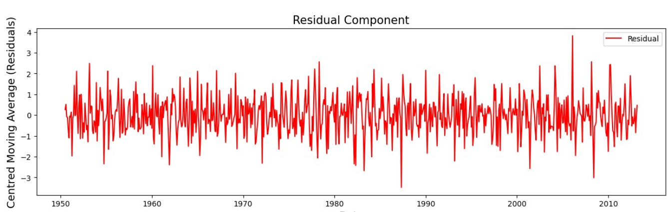

Fig. 16: Residual component in Delhi from 1950 to 2013.

B. Deseasonalization technique

Seasonal patterns in the temperature data were removed using multiple techniques, including:

- Seasonal decomposition library (additive/multiplicative)
- Seasonal differencing
- Estimation of seasonality using local trend

1) Seasonal decomposition library:

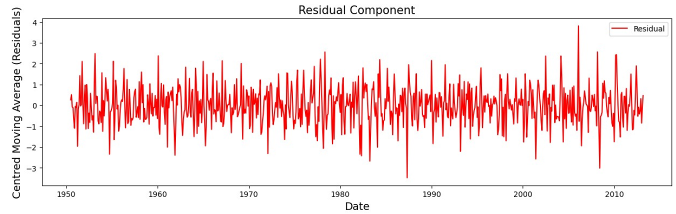

Fig. 17: Residual component in Delhi from 1950 to 2013.

2) Seasonal differencing: Seasonal differencing is used to remove seasonal patterns by subtracting the value from the same season in the previous cycle. The transformation is defined as:

$$y'_t = y_t - y_{t-s} \tag{1}$$

where s is the seasonal period (e.g., $s = 12$ for monthly data).

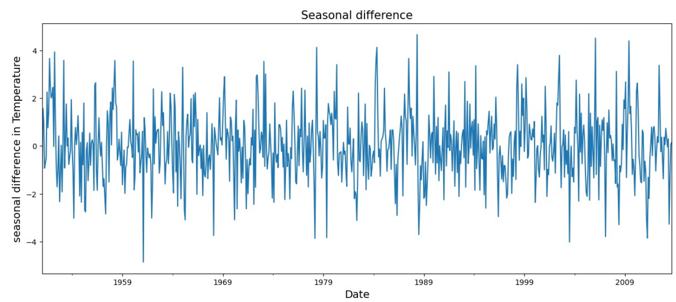

Fig. 18: The plot shows random fluctuations around the mean after removing trend and seasonality. Sharp spikes indicate outliers, possibly due to extreme weather events. The white noise–like behavior confirms effective modeling of the primary components.

3) Estimation of seasonality using local trend :

- **Local Trend Estimation:**

$$\hat{\beta}_j = \frac{1}{k} \sum_{i=1}^{k} x_{ij} \tag{2}$$

- **Row Centered Data:**

$$\xi'_{ij} = x_{ij} - \hat{\beta}_j \tag{3}$$

- **Seasonal Component:**

$$\hat{s}_j = \frac{1}{k} \sum_{i=1}^{k} \xi'_{ij} \tag{4}$$

- **Detrended and Deseasonalized Data:**

$$\eta_{ij} = x_{ij} - \hat{\beta}_j - \hat{s}_j \tag{5}$$

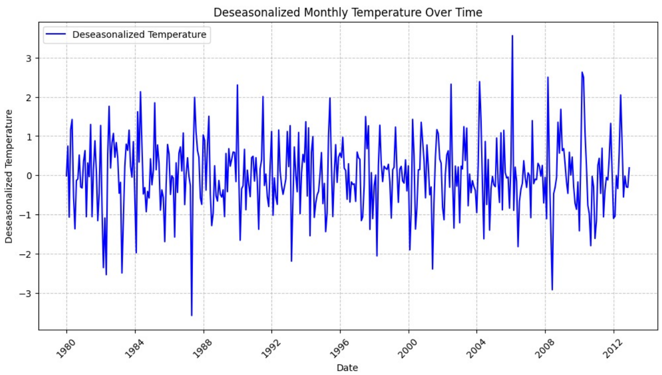

Fig. 19: The plot shows temperature variations with seasonal effects removed, highlighting irregular and trend-related fluctuations overtime.

C. Residual Analysis

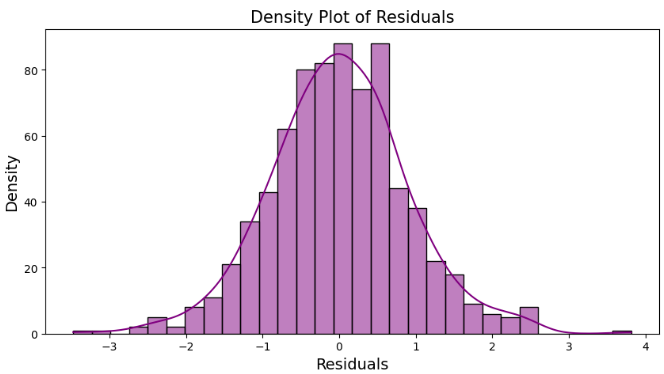

Fig. 20: Density Plot of Residuals from a Regression Model – The density plot shows a roughly normal distribution with a peak around zero, suggesting the residuals are symmetrically distributed and the normality assumption is likely satisfied.

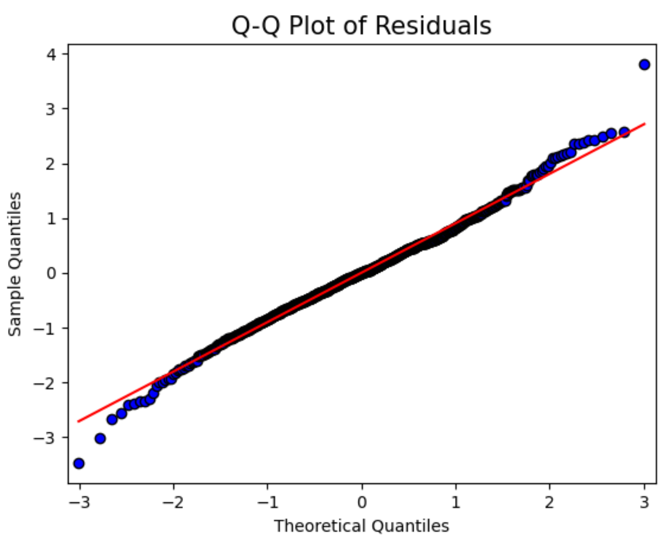

Fig. 21: The Q-Q plot shows the residuals closely follow the red diagonal, indicating they are approximately normally distributed and validating the normality assumption in the regression model.

VIII. Autocorrelation Function

A. ACF and PACF of original Time Series

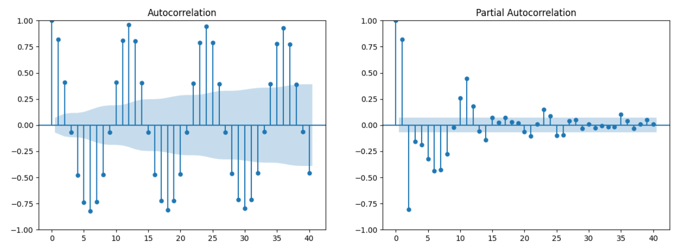

Fig. 22: The plot shows autocorrelation of original data with strong seasonal patterns, significant positive spikes at lags 12, 24, and 36, reflecting annual temperature cycles in Delhi. The consistent autocorrelation across lags indicates a repeating structure typical of seasonal time series.

B. ACF and PACF of residuals

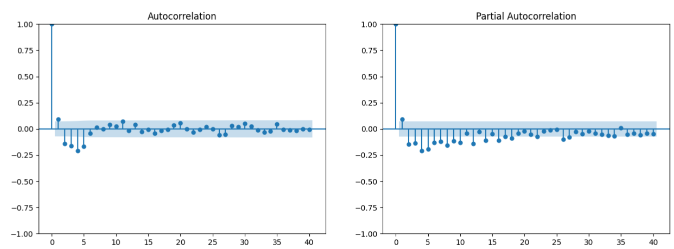

Fig. 23: The autocorrelation plot of the residuals, after deseasonalizing and detrending the surface temperature time series for Delhi (1950–2013), reveals that most autocorrelations lie within the 95% confidence bounds. Except for a few minor fluctuations at early lags, the values hover close to zero, indicating a lack of significant autocorrelation.

The above figure suggests that the primary components of trend and seasonality have been effectively removed. The residuals behave like white noise, implying that the model has captured the underlying structure of the temperature data well. There is no strong remaining pattern in the residuals, supporting the model's adequacy for further forecasting or analysis.

IX. Model Evaluation and Forecasting

A. ARIMA

The ARIMA(12,0,12) and ARIMA(13,0,13) models were chosen for forecasting the monthly average temperature data due to their ability to effectively capture the seasonal patterns observed in the series. Following the standard model selection process, it was determined that the time series was already stationary, so no differencing was required ($d = 0$). The autocorrelation (ACF) and partial autocorrelation (PACF) plots both showed sinusoidal patterns, indicating strong seasonality consistent with a 12-month cycle. By fitting various combinations of p and q, the models with orders (12,12) and (13,13) were found to perform well based on the Akaike Information Criterion (AIC). Residual analysis confirmed the adequacy of the models, as the Q-Q plots showed normally distributed residuals and the ACF of the residuals indicated no significant autocorrelations. Therefore, these models were selected for their robustness in capturing both short-term and seasonal dependencies in the temperature data.

1) Residual Analysis:

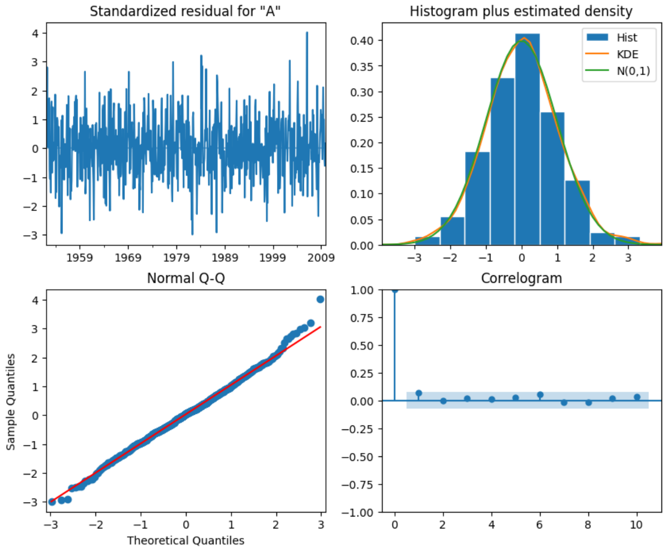

Fig. 24: Diagnostic plots for the ARIMA(12,0,12) model. Residuals show no clear pattern, approximate normality, and lack of autocorrelation, indicating model adequacy.

2) Forecasting using ARIMA model:

Fig. 25: Time series plot showing observed temperatures (blue) and ARIMA forecasts (orange) from 2010, with grey shading for forecast uncertainty. The model captures the seasonal trend well.

3) Comparison with other naive models:

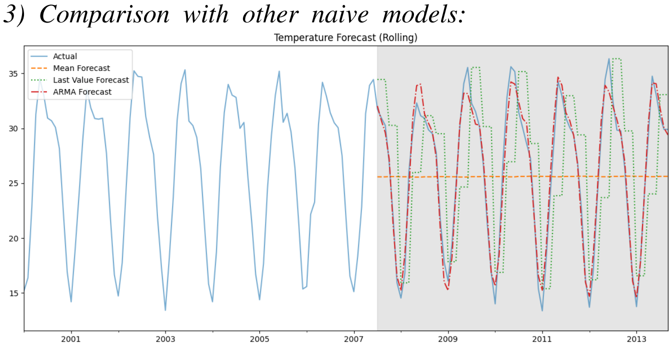

Fig. 26: Temperature forecast comparison using rolling window: ARMA (red) best captures seasonality vs. Mean (orange) and Last Value (green).

4) Model performance evaluation Metrics:

- **Mean Forecast MAE:** 5.8679
- **Last Forecast MAE:** 5.7805
- **ARMA Forecast Metrics:**
  - Mean Absolute Error (MAE): 0.9034
  - Mean Squared Error (MSE): 1.2521
  - Root Mean Squared Error (RMSE): 1.1190
  - Mean Absolute Percentage Error (MAPE): 3.74%
  - Coefficient of Determination (R^2): 0.9713

B. SARIMA

The SARIMA $(2,0,1)$ $(0,1,1)_{12}$ and SARIMA $(1,0,0)$ $(1,0,1)_{12}$ models were selected for forecasting monthly average temperature data due to their explicit capacity to handle both non-seasonal and seasonal components in time series data. The analysis of the original series showed strong evidence of seasonal variation with a 12-month cycle, as confirmed by the autocorrelation and partial autocorrelation plots, which exhibited significant lags at multiples of 12.

The SARIMA$(2,0,1)(0,1,1)12$ model used first-order seasonal differencing to address seasonal trends, effectively capturing both seasonal and short-term dependencies. The SARIMA$(1,0,0)(1,0,1)12$ model directly modeled seasonal autocorrelations without differencing, as the series was already stationary. Model selection was guided by AIC/BIC, and diagnostics confirmed both models were adequate with well-behaved residuals.

1) Residual Analysis:

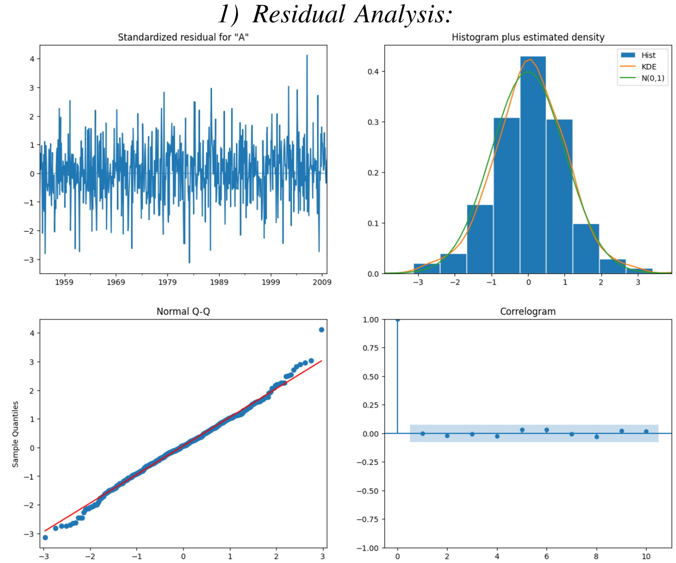

Fig. 27: Diagnostic plots for the SARIMA$(2,0,1)(0,1,1)_{12}$ model. The residuals show no strong patterns or autocorrelations and closely follow a normal distribution, indicating that the model fits the data well.

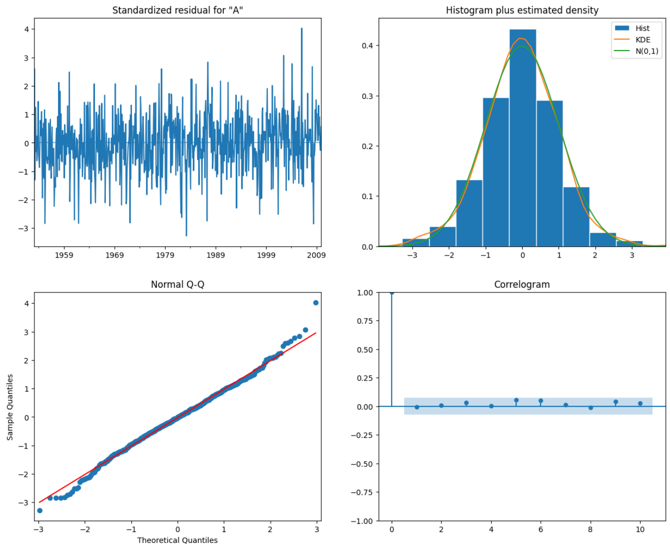

Fig. 28: Diagnostic plots for the SARIMA$(1,0,0)(1,0,1)_{12}$ model. The residuals are well-behaved, showing no strong deviations from white noise or normality assumptions.

2) Forecasting using SARIMA models:

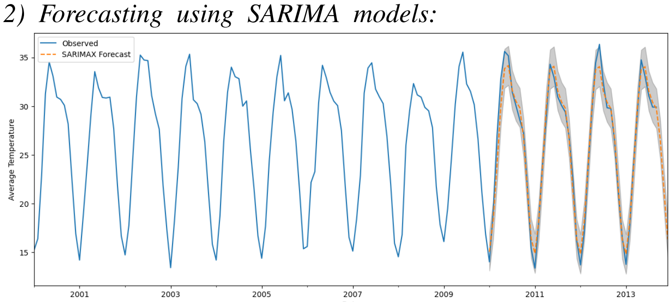

Fig. 29: Forecast from the SARIMA$(2,0,1)(0,1,1)_{12}$ model showing observed temperatures (solid blue) and forecasted values (dashed orange) from 2010 onward. The shaded grey region denotes 95% confidence intervals, and the model effectively tracks the seasonality and trend in the data.

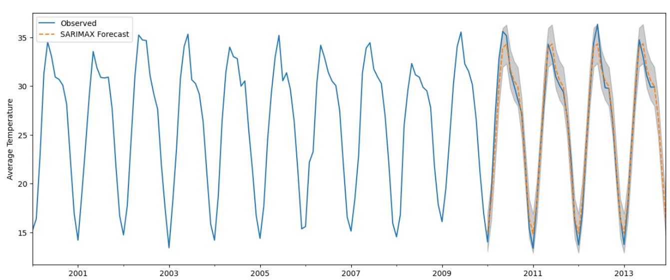

Fig. 30: Forecast from the SARIMA$(1,0,0)(1,0,1)_{12}$ model. Time series plot showing observed average temperatures (solid blue line) and SARIMA model forecasts (dashed orange line) from 2010 onwards. The shaded grey area represents the forecast confidence intervals, indicating the uncertainty associated with the predictions. The model effectively captures the seasonal trend present in the historical temperature data.

3) Comparison with naive models:

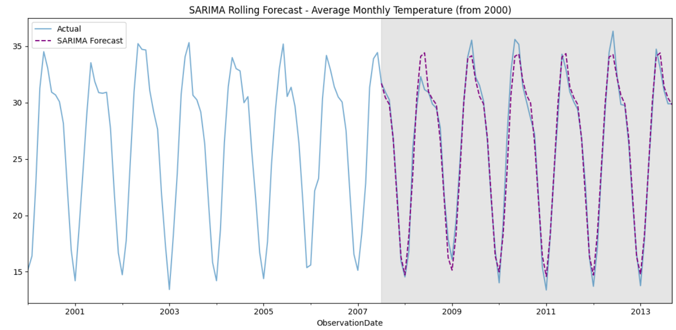

Fig. 31: Comparison of SARIMA forecasts with naive models using a rolling window approach. The plot includes actual temperatures (blue), Mean Forecast (orange dashed), Last Value Forecast (green dotted), and SARIMA Forecasts (red dash-dot).

4) Model performance evaluation metrics:

- **Mean Forecast MAE:** 5.8679
- **Last Forecast MAE:** 5.7805
- **SARIMA**$(2,0,1)(0,1,1)_{12}$ **Forecast Metrics:**
  - Mean Absolute Error (MAE):0.7675
  - Mean Squared Error (MSE): 1.0402
  - Root Mean Squared Error (RMSE): 1.0199
  - Mean Absolute Percentage Error (MAPE): 3.16%
  - Coefficient of Determination (R^2): 0.9781
- **SARIMA**$(1,0,0)(1,0,1)_{12}$ **Forecast Metrics:**
  - Mean Absolute Error (MAE): 0.7565
  - Mean Squared Error (MSE): 1.0128
  - Root Mean Squared Error (RMSE): 1.0064
  - Mean Absolute Percentage Error (MAPE): 3.1120%
  - Coefficient of Determination (R^2): 0.9767

X. The Need for Deep Learning Models

Though the ARIMA and SARIMA models are good at predicting the future values based on training and testing the datasets, they come with their limitations.

Taking their time in figuring out the p, q, d values, they often run for longer durations. For larger datasets, especially the ones with more than 10,000 data points, the code runs endlessly. Furthermore, the SARIMAX model in not capable of capturing more than one seasonality. For example, for the temperatures recorded hourly for 10 years, there are going to be two sets of seasonality. The temperature would be higher in the afternoon compared to the morning and night. Also, the temperature would be higher in the summers compared to the winters.

These challenges cause us to go further ahead, beyond just SARIMAX models, to implement the Deep Learning Models. They use hidden layers and neurons in order to estimate the relationship between the input and the output, reducing the processing time as well as the errors.

XI. Deep Learning Models – An Introduction

There are three main types of deep learning models that we can build for time series forecasting: Single-step models, multi-step models and multi-output models.

The single-step models focus on one variable, one step into the future. The multi-step models however, focus more than one step into the future, but just for one variable. The multi-output models, which won't be required in our implementation of the dataset, is similar to the multi-step model, but focuses on more than just one variable.

We would aim to implement deep learning models having both linear and non-linear relationship between the input and the output. This would be done by incrementally adding hidden layers in the network. We would then proceed to implement the sequential models, namely the RNN model, and its specific types, LSTM and GRU.

A. RNN Model

It is a type of neural network designed to handle sequential data, where current output depends on previous inputs — like time series, text, audio, etc.

It has a loop that lets information persist over time, making it different from normal feedforward networks. However, it struggles with long-term dependencies and suffers from vanishing gradients during training.

B. LSTM Model

It is an advanced type of RNN that can remember information for a long time — designed specifically to fix the weaknesses of standard RNNs. Uses gates, namely the forget gate, input gate and output gate to control memory flow. It is very good at long-term dependencies as it maintains a current state of the system along with the output and works well for long sequences like language, weather, or stock forecasting.

C. GRU Model

It is a simplified version of LSTM that performs similarly but is faster to train. This is because it has only 2 gates, namely the reset gate and the update gate. It therefore, has far fewer parameters than LSTM, and is used when there is a requirement of speed and low complexity.

XII. Deep Learning Models – Implementation

For the single-step and the multi-step models, we will train and test our data for different implementations. Firstly, we would figure out a baseline comparison, which would include the last value read and the last window read. These baseline models would help us evaluate how efficient our deep learning models are compared to standard implementation techniques.

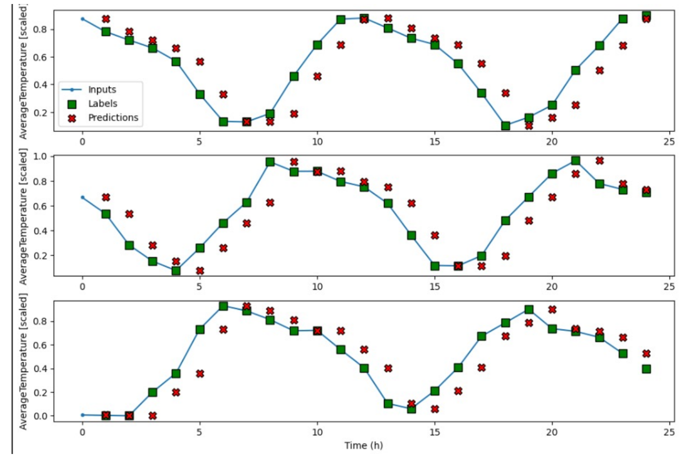

Fig. 32: Single Step Base line model

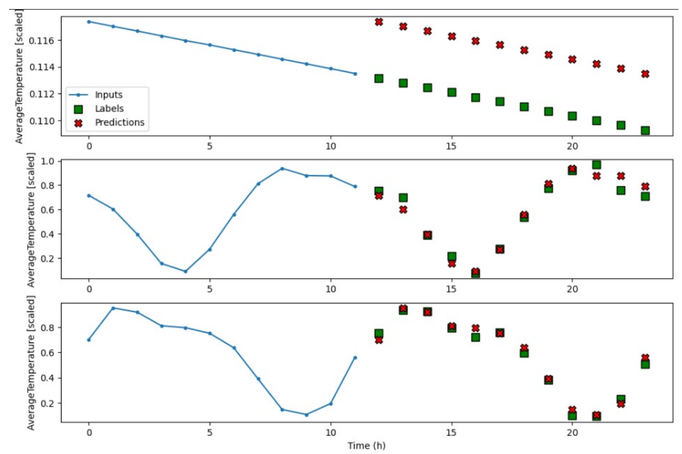

Fig. 33: Multi step base line model - Repeat last 24 values

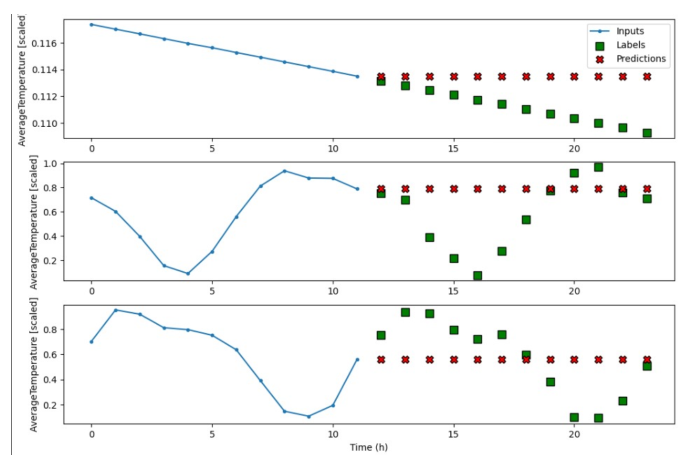

Fig. 34: Multi step base line model - Repeat only last value

Secondly, we would then proceed to the linear models, where the relationship between the input and the output shall be a linear function. This would be similar to the linear regression that we have studies so far.

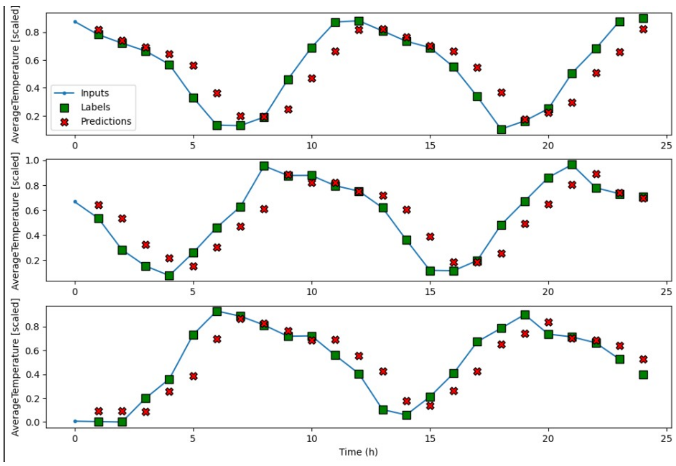

Fig. 35: Single step linear model

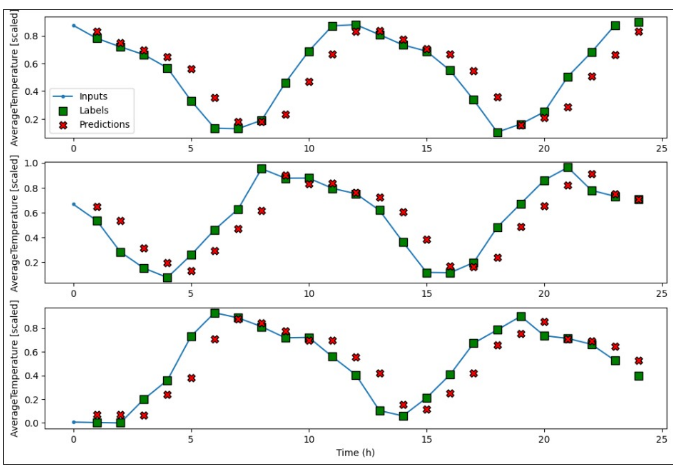

Fig. 36: Multi step linear model

Next, we would proceed to add non-linearity in the relationship of the input and output, using hidden layers with non-linear activation functions such as ReLu.

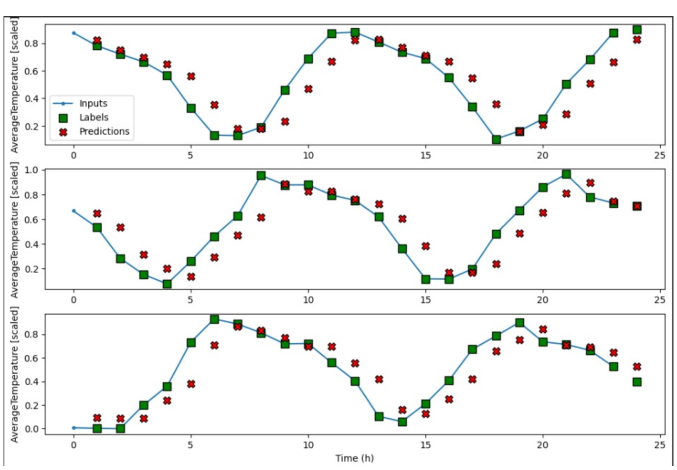

Fig. 37: Single step dense neural network

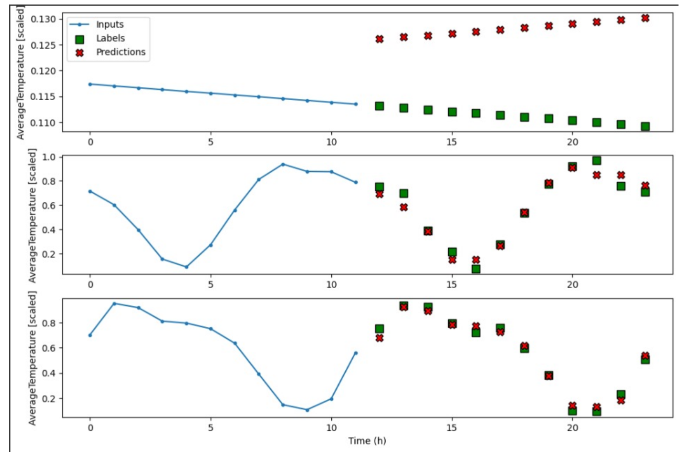

Fig. 38: Multi step dense neural network

With the deep learning models now being implemented, we could observe the MAE values reduce as we go further. Finally, we would implement the sequence models of deep learning, RNN, LSTM and GRU.

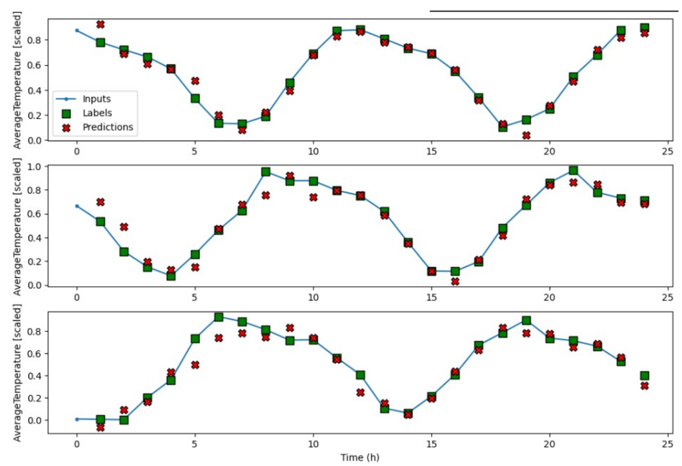

Fig. 39: Single step RNN

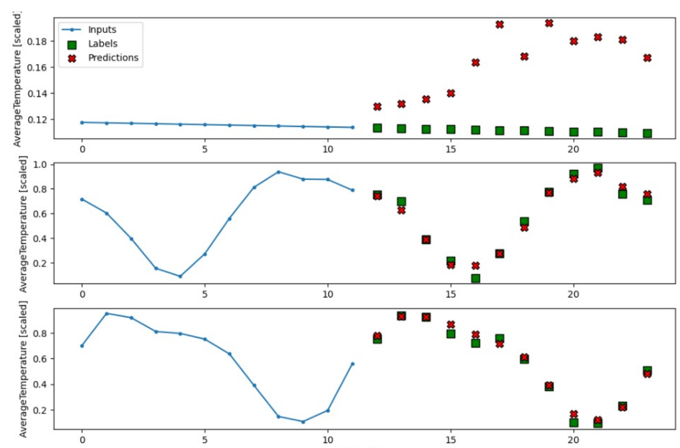

Fig. 40: Multi step RNN

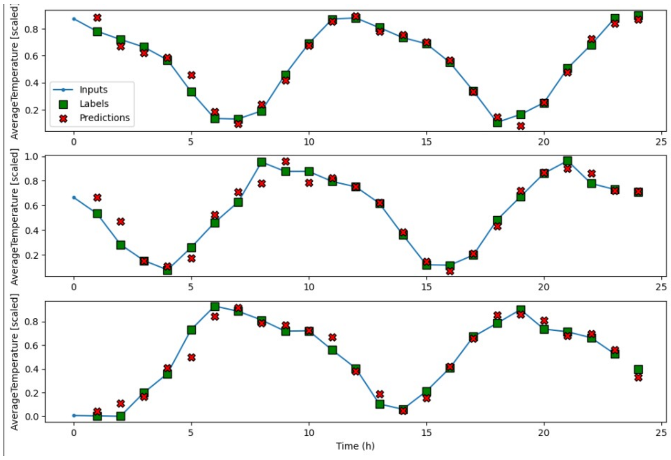

Fig. 41: Single step LSTM model

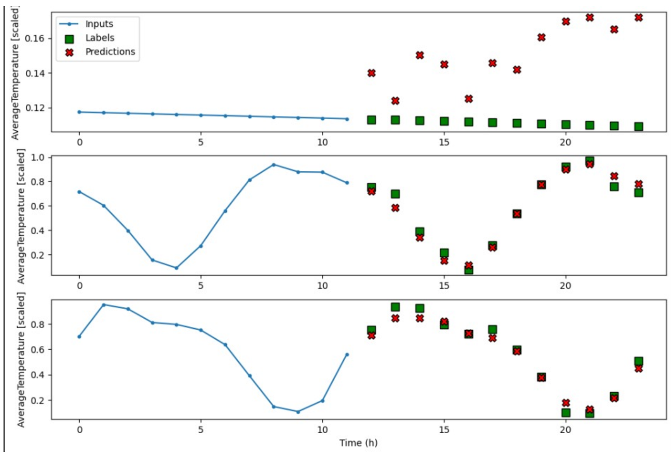

Fig. 42: Multi step LSTM model

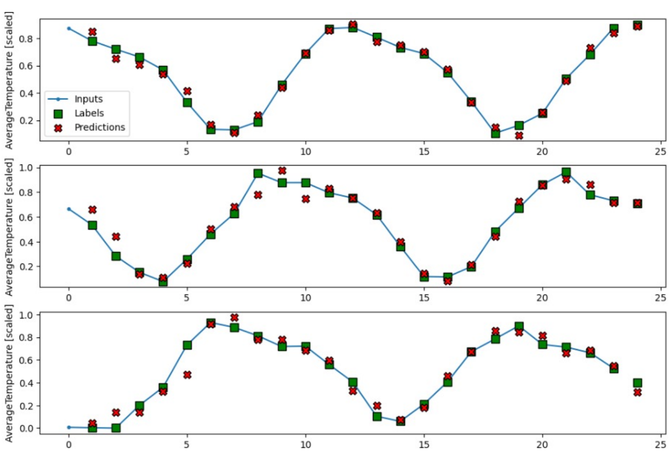

Fig. 43: Single step GRU model

Keeping MSE, Mean Squared Error as the loss function and MAE, the Mean Absolute Error as the metric for comparison, we can observe how the prediction improves and the prediction and labels overlap. The GRU model, being a more efficient and faster implement of the LSTM model, gives the output faster, but, the resulting MAE is larger, due to the assumptions taken in the process.

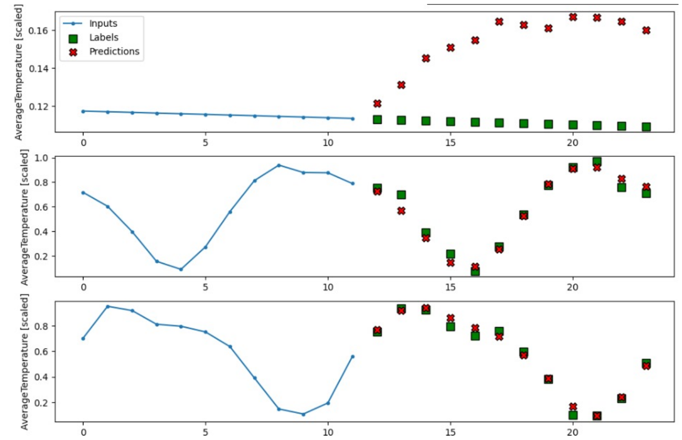

Fig. 44: Multi step GRU model

XIII. Deep Learning Models – Result

From the implementation, we could see the better results that are offered by the Deep Learning Models. Though the linear model is a far better improvement than our baseline models, the improvement can be further seen in the RNN models and non-linear models. The performance of the LSTM, RNN and GRU models is further improved compared to the non-linear models. The LSTM model gives the best MAE result as the LSTM has a longer memory than the RNN. The GRU cannot give a lower MAE, as it trades off the prediction accuracy for faster output.

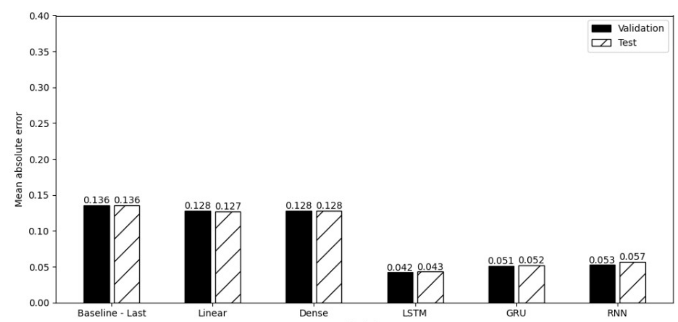

Fig. 45: Single step model performance

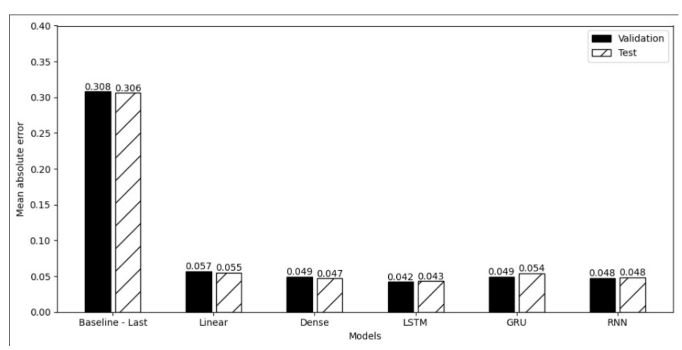

Fig. 46: Multi step model performance

XIV. CONCLUSION

The study concluded a clear long-term warming trend from the 1980s onward, along with increased temperature variability. Seasonal patterns remained consistent, with peak temperatures in May–June and lows in December–January. Statistical measures like skewness and kurtosis provided deeper insights into temperature distribution, highlighting the prevalence of warm anomalies and changes in extreme events. Deseasonalization techniques, such as seasonal decomposition and differencing, effectively removed cyclic patterns to expose underlying trends. ARIMA and SARIMA models effectively captured both seasonal and non-seasonal patterns in the data. Residual diagnostics showed minimal autocorrelation and approximate normality, confirming model adequacy. Among the two, the SARIMA model demonstrated better performance with lower forecasting error. Overall, the study demonstrated the value of time series analysis in climate research and forecasting, offering meaningful insights into Delhi's changing climate and showcasing robust methods to analyze long-term environmental trends.

However, despite their strengths, traditional statistical models like SARIMA face notable limitations in handling large datasets and complex seasonal patterns—particularly when multiple seasonalities are present. This necessitates the exploration of deep learning models, which offer greater flexibility and efficiency. Models like RNN, LSTM, and GRU leverage sequential learning and internal memory to capture intricate temporal dependencies and nonlinear relationships in the data. Their implementation showed promising improvements in predictive accuracy, especially when non-linear functions and hidden layers were introduced. While baseline and linear models offered valuable benchmarks, the deep learning models, even on modest datasets, outperformed them in terms of error metrics like MAE. These findings reinforce the growing importance of integrating deep learning approaches in climate forecasting, particularly as the scale and complexity of environmental data continue to increase.

REFERENCES

[1] Kaggle, *Climate Change: Earth Surface Temperature Data*

[2] Hindustan Times, *Delhi: February this year second warmest since 1901, says IMD*

[3] Business Standard, *Warmest night ever recorded in Delhi, 5 dead from heatwave: Details here*

[4] India Today, *Delhi records warmest October in 74 years with maximum temperature of 35.1°C*

[5] The Times Of India, *2008 witnessed warmest December*

[6] Time Series Forecatsing in Python, *Marco Peixeiro*