UNIVERSIDADE D COIMBRA

Weather Time Series Analysis and Forecast

Time Series Analysis and Pediction

Master in Computer Science Engineering
2024/2025

December 10, 2024

Authors

João Moreira

joaomoreira@student.dei.uc.pt

PL1

2020230563

Tomás Pinto

tomaspinto@student.dei.uc.pt

PL1

2020224069

Contents

1. Introduction

1.1. Problem description

The analysis and forecasting of weather patterns play a significant role in a variety of fields, from agriculture to energy management and infrastructure planning. By studying historical weather data, we can identify trends and seasonal variations that help us better understand local weather conditions and predict future patterns.

This project focuses on the time series analysis of mean temperature data for a selected location. The goal is to model and forecast temperature trends using different time series analysis techniques, including exponential smoothing and other machine-learning methods, to assess their predictive accuracy.

In addition to forecasting mean temperature, we extend the analysis by incorporating multivariate models that consider other weather variables. This approach allows for a more comprehensive understanding of the relationships between different weather factors, ultimately leading to more reliable forecasts.

1.2. Dataset description

The dataset used for this project was obtained from the Open-Meteo Historical Weather API, which provides daily weather observations for various locations worldwide. For this analysis, we selected data from *Tokyo* for the period spanning from January 1st, 1940, to the present. The primary variable of interest is the mean daily temperature, supplemented with additional weather variables for multivariate analysis.

The dataset contains daily observations structured with a date column and corresponding values for each weather variable. An initial inspection of the dataset revealed some missing entries, in the first day. We opted to remove the first day of the dataset because it does not impact the time dependency of the time series.

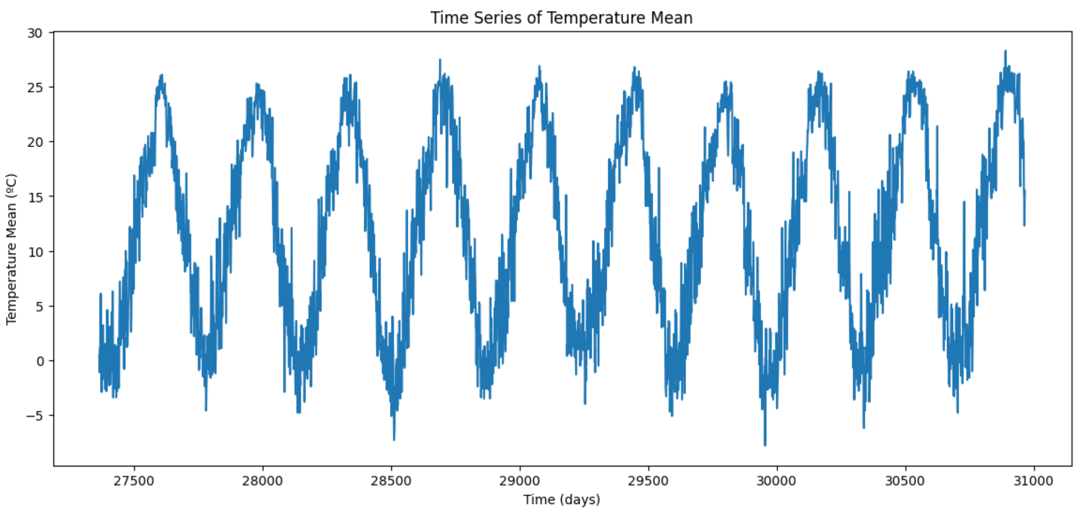


Figure 1: Visualization of mean temperature TS

In order to generate the figure 1 and facilitate visualization, we plotted the mean temperature values for each day over the last 10 years of recorded data. First things first, there are some aspects to notice in our mean temperature time series:

- There is a very small, almost unnoticeable trend.
- The time series is remarkable because of the constant seasonal pattern.
- The seasonal pattern does not change with the trend.

Based on these observations, the additive model is appropriate. In this model, the three components are independent, and it is usually used when the magnitude of seasonality and noise do not change with the trend, which aligns with the characteristics observed in the time series:

$$x(n) = tr(n) + sn(n) + e(n)$$

As follows, this time series is not stationary at all. In chapters **TO REF** we will see more about this topic.

The last thing to point out is that the first 80% of the dataset will be used for characterisation, modelling, and training and the last 20% will be used to assess the forecasting performance of our models.

2. Decomposition Methods

In this section, we present the TS decomposing methods in its main components:

- Trend: Captures the long-term progression in the series. It represents the overall direction in which the data is moving (upward, downward, or stationary) over time.
- Seasonality: Refers to patterns that repeat at regular intervals, often driven by external factors such as seasons of the year.
- Erratic Components: Random variation in the data that cannot be explained by the trend or seasonality. It represents unpredictable fluctuations that might arise from various short-term factors.

2.1. Model Fitting Approaches

Model fitting is a corrective trend method, which involves selecting the right mathematical function or statistical model to capture the trend and seasonality. One common method is *Polynomial Fitting*, where the theory says that a TS can be described as a polynomial as follows:

$$T(n) = a_0 + a_1 n + a_2 n^2 + ... + a_t n^t$$

where $a_0, a_1, ..., a_n$ are the polynomial coefficients, and t is the time variable. For instance, a higher-degree polynomial might capture more complex trend patterns, but it also risks overfitting the data.

2.2. Local Smoothing Approaches

Local smoothing techniques are methods used to extract the trend by averaging over a small, sliding window of the data. Those involve averaging or weighting nearby data points to smooth the series.

2.2.1. Moving Average (MA)

The moving average is a simple and effective technique for smoothing. It calculates the average of a fixed number of neighboring points (a window) and replaces each point with its average:

$$T(n) = \frac{1}{\sum \omega_k} \sum_{k=-\frac{M-1}{2}}^{\frac{M-1}{2}} \omega_k x(n+k)$$

beign $\omega = [-\frac{M-1}{2}, ..., \frac{M-1}{2}]$ the weights of the filter.

2.2.2. LOWESS/LOESS (Locally Weighted Scatterplot Smoothing)

LOWESS is a more flexible, non-parametric method for smoothing. It fits multiple regressions to local data segments, weighting each point based on its distance from the current data point. This approach is ideal for capturing non-linear trends without assuming a specific form for the trend component.

Unlike moving average, which uses a fixed window, LOWESS adjusts its fit based on the local structure of the data, making it more adaptable in the presence of complex, non-linear trends.

2.3. Seasonality assessment

Identifying and measuring seasonality is crucial for understanding the recurring behaviors in the data, which helps in forecasting and making data-driven decisions.

Two common techniques for assessing seasonality are Filtering and Epoch Averaging.

2.3.1. Filtering

Filtering is a technique used to isolate the seasonal component of a time series by removing trend and noise. Common methods include the already presented moving average filter (low-pass), with a window size often chosen based on the seasonal period, and Fourier transform filtering, which identifies and removes specific frequency components corresponding to the seasonal cycle.

2.3.2. Epoch Averaging

Epoch averaging, also known as seasonal averaging, involves breaking the time series into repetitive cycles (epochs) and averaging the corresponding points from each cycle. For instance, averaging the values for the same month across different years can reveal the seasonal pattern. This method is useful for highlighting consistent seasonal trends by smoothing out random fluctuations.

2.4. Trend and Seasonality removal by differencing

Differencing is a key technique used to remove trend and seasonality from a time series, transforming a non-stationary series into a stationary one.

2.4.1. First-order differencing

This method works by subtracting the value of the current observation from the previous one.

$$\Delta x(n) = x(n) - x(n-1)$$

This operation removes trends or long-term fluctuations in the data, highlighting short-term changes and making the series more stable in terms of its mean and variance. First-order differencing is particularly useful when the series exhibits a linear trend.

2.4.2. First-order seasonal differencing

This approach targets seasonal patterns, which occur at regular intervals. Instead of subtracting consecutive points as the previous technique, it subtracts values from the same season in previous cycles.

$$\Delta x(n) = x(n) - x(n-s)$$

3. Stationary Assessment Methods

Once both trend and seasonality have been removed, we assess the series's stationarity. Ensuring stationarity is essential, as non-stationary data can complicate forecasting models. We then evaluate the series' autocorrelation by constructing correlograms, which illustrate the time-dependent relationships within the data. Finally, we conduct a formal stationarity assessment using statistical tests.

3.1. Autocorrelation and Correlogram

Autocorrelation is a statistical measure that describes how the current value in a time series is related to its past values. In other words, it evaluates the degree of similarity between observations as a function of the time lag between them. The Autocorrelation Sequence (ACS) calculates these correlations for different time lags, allowing us to identify patterns, such as periodicity or seasonality, and understand the dependencies between observations over time.

The confidence bounds on the correlogram are critical for interpreting the results. These bounds indicate the range of values within which we expect the autocorrelations to lie if the data is random (i.e., if there's no meaningful autocorrelation at that lag). If the autocorrelation values fall outside the confidence bounds, it suggests that there is significant correlation at that specific lag, which could indicate a strong seasonal effect or trend.

3.2. Visual Approach

A visual inspection of the time series plot is an intuitive method to assess stationarity. By examining the graph, one can look for a constant mean and variance over time. If the series shows no noticeable trends or long-term shifts in behaviour, it might be considered stationary. However, trends or periodic patterns would indicate non-stationarity, requiring further transformation, such as differencing, to stabilize the series.

3.3. Statistical Approach (Unit Root Test)

For a formal stationarity check, statistical tests like the Augmented Dickey-Fuller (ADF) test are applied. The ADF test evaluates the null hypothesis that the series contains a unit root, which implies non-stationarity. If the test rejects the null hypothesis, it indicates that the series is stationary. This is a crucial step before applying any time series modeling techniques, as many models assume the series is stationary for accurate forecasting.

4. Time Series Forecasting

Forecasting involves predicting future values based on historical data and observed patterns. Time series forecasting techniques vary in complexity and applicability, depending on the nature of the data and the specific forecasting needs. Below, we discuss key methods and concepts in forecasting.

4.1. Box-Jenkins Forecasting

The Box-Jenkins methodology focuses on identifying, fitting, and validating ARIMA and SARIMA models for forecasting. It emphasizes three iterative steps: model identification (choosing AR, MA, and differencing orders), parameter estimation, and diagnostic checking. ARIMA models are particularly suited for stationary series and account for both autocorrelation and noise, making them versatile for many forecasting applications.

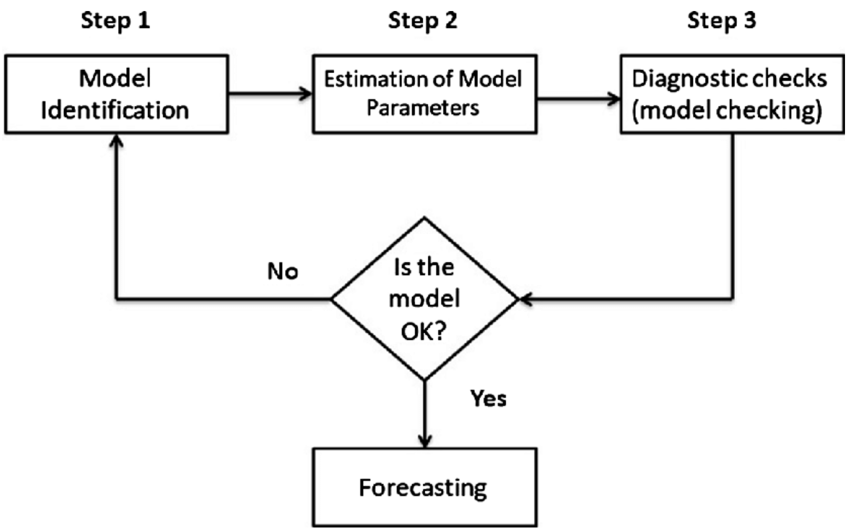


Figure 2: Box-Jenkins forecasting diagram

4.2. Evaluation

To assess the performance of our developed models, we employ several widely used evaluation metrics: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). These metrics provide insights into the accuracy of the predictions by quantifying the difference between the predicted and actual values.

When the goal is to identify the best model, we also consider the Akaike Information Criterion (AIC). This criterion strikes a balance between the goodness of fit and model complexity, helping to prevent both overfitting and underfitting. By incorporating a penalty for excessive complexity, AIC ensures that the selected model is both accurate and parsimonious.

4.2.1. Preprocessing

We decided to aggregate the original daily data into monthly averages. This transformation simplifies the dataset while retaining the key seasonal and trend components crucial for forecasting mean temperatures.

Additionally, we opted to focus on the last 10 years of data. This decision was based on the observation that a significantly larger dataset spanning multiple decades introduced unnecessary computational complexity and diminishing relevance for predicting near-term trends. By limiting the dataset to the most recent decade, we ensured that the model captured current seasonal and trend dynamics effectively.

After this preprocessing step, we divided the time series into a training set (the data excluding the last 24 months), a validation set (from the last 24 months to the last 12 months), and a test set (the last 12 months):

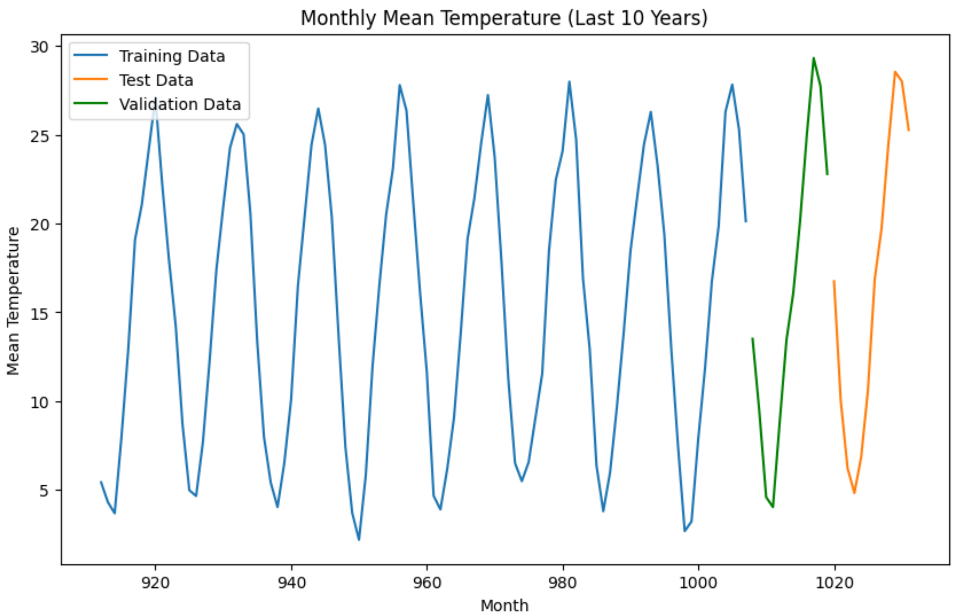


Figure 3: Visualization of mean temperature TS

4.3. Estrapolation of trend curves

Extrapolation involves extending observed trends into the future using fitted trend models. Common approaches include polynomial fitting, linear regression, or more complex nonlinear models. The simplicity of extrapolation makes it appealing for data with well-defined trends, but it assumes that past trends will persist, which may not always hold true in dynamic environments.

4.4. Exponential Smoothing (ES)

Exponential Smoothing is a family of extrapolation methods that predict future values by weighting past observations, with recent data receiving more emphasis. It adapts well to time series with different levels of complexity and is widely used for short-term forecasts. This type of forecasting is proper when TS is formed by samples that summarize periodic data, for example, one sample per year, summarizing all year.

4.4.1. Triple Exponential Smoothing (TES)

Triple Exponential Smoothing (TES), also known as the Holt-Winters method, is an advanced forecasting technique that accounts for seasonality, trends, and level in a time series. It extends methods like Simple Exponential Smoothing (SES) and Double Exponential Smoothing (DES) by introducing a third component to model seasonal fluctuations. TES can handle additive or multiplicative seasonality, depending on the characteristics of the series. This method seems to be the most promising of the three for our case.

4.5. Prediction Intervals

Prediction intervals provide a range within which future values are likely to fall, accounting for the uncertainty inherent in forecasting. They are derived from the residuals of the model and help quantify the confidence in the predictions. Wider intervals indicate higher uncertainty, while narrower ones suggest more reliable forecasts.

4.6. Sarima

In this project, we focus exclusively on SARIMA for our forecasting tasks. Traditional models such as AR, MA, ARMA, ARIMA and ARFIMA are not utilized independently, as their functionalities are inherently integrated into SARIMA. By leveraging SARIMA, we capture both the seasonal and non-seasonal dynamics of the data within a unified framework, streamlining the modeling process while addressing the complexities of seasonal trends.

This approach ensures a comprehensive and efficient modeling process, aligning with the structured methodology of Box-Jenkins.

5. Time Series Selection

To select the most relevant TS for forecasting the mean temperature, we will construct a cross-correlogram between the mean temperature and the other TS in the dataset. This analysis will help identify the strength and significance of lagged correlations between the target variable and potential predictor series. By evaluating the cross-correlation at different lags, we can determine which TS exhibit the strongest predictive relationship with the mean temperature, guiding the selection of features for model development. This approach ensures that only the most informative time series are used, improving the accuracy and efficiency of the forecasting process.

6. Multivariate modelling and Forecasting

6.1. Vector Autoregressive Model

The VAR model captures the linear interdependencies among multiple variables by treating each variable as a function of its own past values and the past values of other variables in the system. The VAR model is particularly suited for understanding dynamic relationships and making forecasts in systems where variables influence each other over time. We made sure that the data were stationary before applying the model.

6.2. SARIMAX Model

The SARIMAX (Seasonal Autoregressive Integrated Moving Average with Exogenous Variables) model is well-suited for capturing seasonal patterns, trends, and dependencies on external variables. The data preparation involved grouping daily observations into monthly averages and differencing to ensure stationarity. The SARIMAX model was configured with both autoregressive (p) and moving average (q) components, seasonal terms (P, Q, s), and integrated differencing (d, D) based on exploratory data analysis. Each variable was modeled as the endogenous series, while the remaining variables served as exogenous predictors. The model was trained on 10 years of historical data, and its performance was evaluated using one-year-ahead forecasts compared to unseen test data.

6.3. Multilayer Perceptron Model

A MLP regressor was implemented to predict values. The input data was preprocessed using a sliding window technique to create features and labels, followed by normalization with the StandardScaler to ensure a mean of 0 and a standard deviation of 1. The model was trained using the Adam optimizer with a mean squared error (MSE) loss function, and its performance was evaluated using the same metrics talked about before. This approach provides a solid foundation for analyzing and forecasting time series data as it was done in previous courses, as pattern recognition.

7. Assessment Results

This section presents the outcomes of various time series analysis techniques applied to model and understand the trend, seasonality, and noise within the dataset. The focus is on determining the most suitable techniques for achieving stationarity in the time series, ultimately supporting robust forecasting.

7.1. DFT Computation

To evaluate the presence of seasonal cycles in the data, we applied a Discrete Fourier Transform (DFT) to the detrended time series data. This approach allowed us to analyze the frequency components by converting the time series from the time domain to the frequency domain.

The power spectrum of the transformed data, computed as the squared magnitude of the Fourier coefficients, is presented in Figure 4. The sampling frequency was set to 365 samples per year, reflecting daily observations.

The power spectrum reveals a distinct spike at a frequency of 1 cycle per year, indicating a strong annual seasonality in the data. This finding aligns with the expectation of yearly patterns commonly observed in weather-related variables.

7.2. Polynomial Fitting on the series

In this method we tried to fit the trend with different polynomial degrees: 1^{st} order, 2^{nd} order and 20^{th} order.

As the degree of the polynomial increases, the trending tends to overfit the TS. As it is possible to see in figure 5, for a 20^{th} order we get a stationary TS. This happens because we are

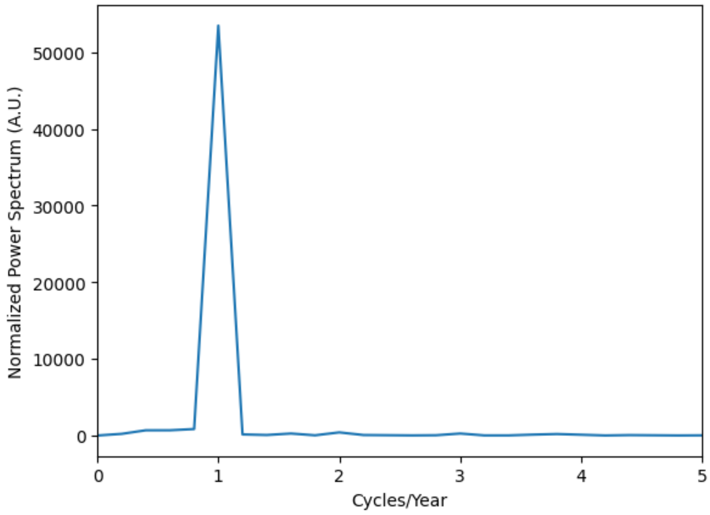
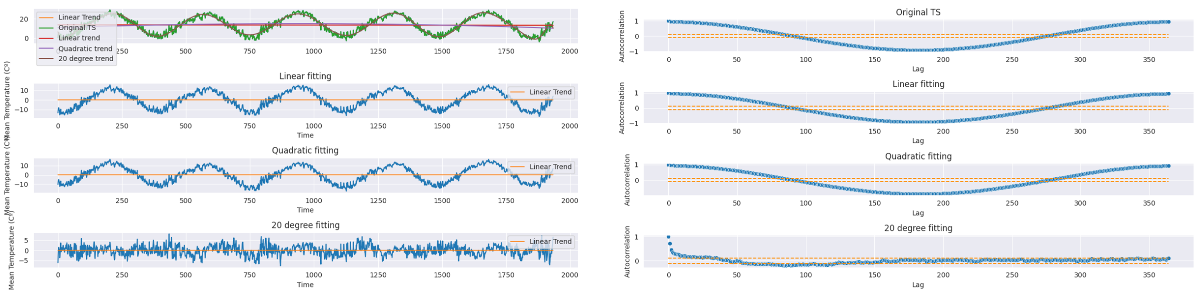


Figure 4: DFT Computation

removing the trend and the stationary component at the same time. The *ADF* test gives us a value of: -8.025942, and we reject the null hypothesis with a 99% confidence degree.



(a) TS with polynomial fitting

(b) Correlogram

Figure 5: Polynomial fitting trend estimation

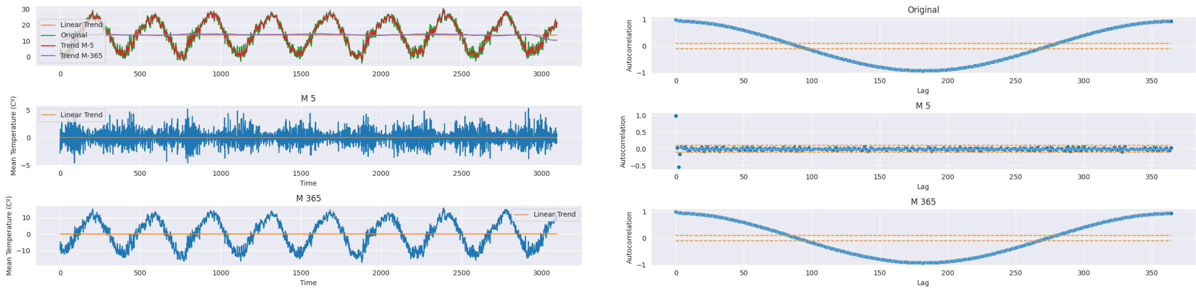
For a 1ª and 2ª order we obtained the values -2.153432 and -2.032724 for *ADF test* respectively, which means the TS is not stationary. This method is not very useful as it does not give us a stationary TS when we try to remove the trend. To remove the stationary from a very long TS we need to increase the polynomial degree to a high number which is very difficult to calculate.

7.3. MA Smoothing

As it is possible to see in figure 6a, for a window of size $M = 5$, the model overfits the TS. We can say this because the trend has completely adjusted to the seasonality curves, and once removed will originate the erratic component. We can confirm this if we look at figure 6b, where the correlogram says that the TS is completely stationary with a *ADF value* of -20.87.

On the other hand, with a M equal to the period of seasonal pattern, $M = 365$, the trend indeed follows the real long trend of TS, and that is why we don't see much of a difference from the original series when applied.

If we want to assess the erratic component we can use a smaller window size, such as $M = 5$, and to assess the seasonal component we can use a window size of $M = 365$.

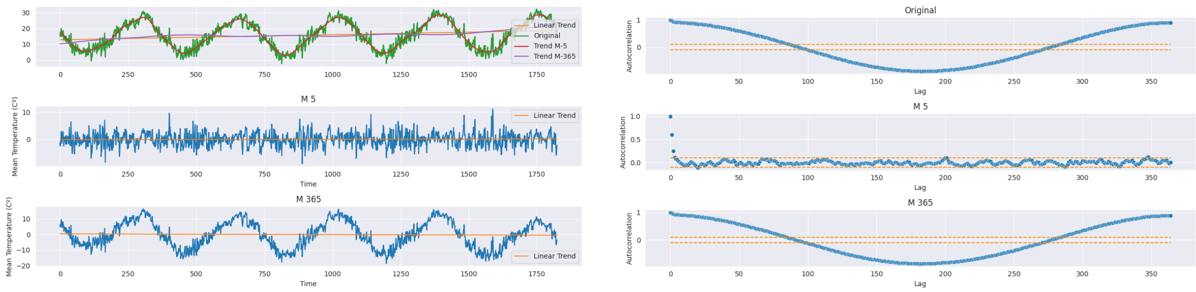


(a) TS with ma smoothing (b) Correlogram

Figure 6: MA Smooth trend estimation

7.4. LOWESS

Our analysis of local smoothing techniques using moving average and LOWESS methods revealed that both methods obtained similar results. Below 7 we present the results for the LOWESS method:



(a) TS with local smoothing (b) Correlogram

Figure 7: Local Smooth trend estimation

As the span of points used, we experimented with 5, a small window, that resulted in a good smooth but kind of overfitting, not perfectly identifying the main long trend. We also used a span of 365, as the yearly seasonality is present, where the trend is identified more precisely.

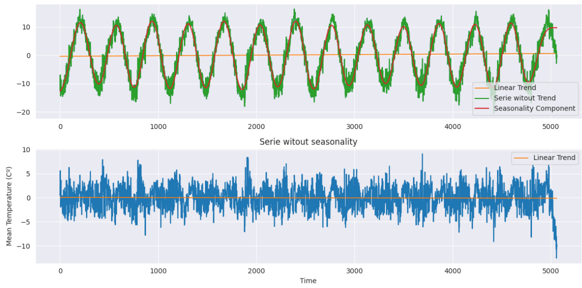
The *ADF statistic* for the window of size 5 is: -9.94 and if we look at the correlograms, the *MA Smoothing* correlogram has more points closer to zero than *Local Smoothing Method.*

In short, we can say *MA Smoothing Method* might work better than *Local Smoothing Method.*

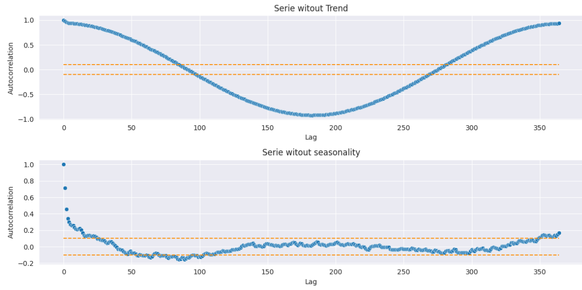
7.5. Filtering

In order to remove the seasonality we applied a low-pass filter to remove the frequencies on TS, we previously used the *MA Smoothing Method* with $M = 365$ to remove the trend. This method was chosen because it is the one that has the lowest *ADF values* in the end of the experiment.

In figure 8a we can see that the seasonality component fits the TS. The correlogram is not perfectly close to zero, nonetheless, we can say the TS is stationary with a confidence bound of 99% for a *ADF value* of -12.03.



(a) TS with Filtering method

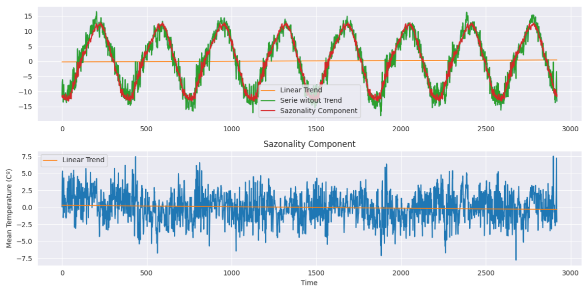


(b) Correlogram

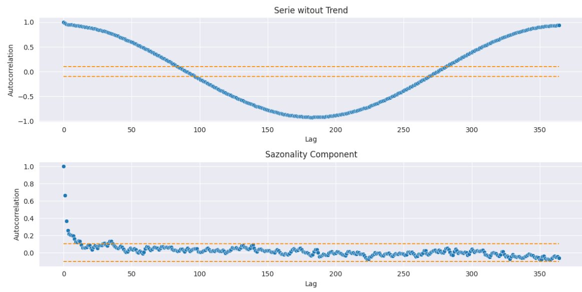
Figure 8: Filtering seasonality estimation

7.6. Epoch Averaging

We also assessed the seasonality using epoch averaging in the same conditions, achieving a similar outcome. In this method, we defined the seasonal cycle to be 365 days, as the yearly periodicity of the data was evident. By averaging the values over each cycle, we obtained an estimate of the seasonality, which we then removed to isolate the erratic component of the time series. We can see these results in figure 9a.



(a) TS with epoch averaging method



(b) Correlogram

Figure 9: epoch averaging seasonality estimation

Similarly to filtering, the correlogram is not perfectly close to zero, but we can say that the TS is stationary with a confidence bound of 99% for a *ADF value* of -14.46.

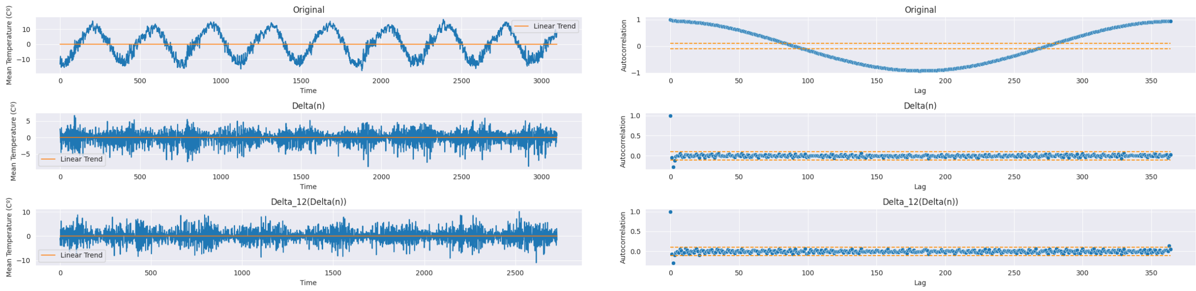
The low-pass filter approach provided a smoother result, while the epoch averaging method fitted a bit more. In the end, both retained the same overall outline.

The remaining irregular fluctuations, in bottom figures 9a and previous 8a, represent the random noise and short-term variations, now more visible without much trend and seasonality influence.

7.7. Trend and Seasonality removal by differencing

To apply differencing, we first took a one-step difference, followed by a second difference with a lag equal to the frequency rate, based on the previously differenced values.

As we can see in figure 10a, when we applied the first order differencing, it was expected by us to remove the trend but at the same time we removed the seasonality too. This behaviour already happened before and occurred again. In the correlogram is visible that the TS is stationary, with a *ADF value* of -7.49.



(a) TS with differencing method (b) Correlogram

Figure 10: Filtering trend and seasonality estimation

We notice that this method outputted a similar result to *MA smoothing method* for a $M = 5$. This probably happened because the samples are too close to one another and doing the difference results in a similar value for the mean of the neighbours.

8. Data Series Univariate Forecasting Results

In this section, we present the results of our time series forecasting efforts, focusing on predicting monthly mean temperatures.

8.1. SARIMA

To estimate the parameters for the *SARIMA Model*, we calculate the *ACF* and *PACF* for the time series (TS) after removing trend and seasonality. The results are shown in Figure 11.

Empirically, we know the seasonality is 12 months, i.e., $s = 12$. Since our time series is non-stationary and exhibits seasonality, we estimate d and D as 1. By examining the first 12 lags of the ACF and PACF, we identify the values of q and p as 3 and 5, respectively. To determine Q and P, we focus on the lags that are multiples of 12, resulting in values of 1 and 2, respectively.

In figure 12 it is possible to visualize our forecast for the last year for SARIMA((5, 1, 3), (2, 1, 1), 12).

It is a satisfactory result as the prediction almost covers the true values. We can see the test results on table 1:

Measure	Value
MSE	1.84
RMSE	1.36
MAE	1.11
MAPE	10.61

Table 1: Forecast measures of SARIMA((5, 1, 3), (2, 1, 1), 12)

8.1.1. Grid Search

To identify the best SARIMA model, we implemented a grid search algorithm using the parameter ranges detailed in Table 2.

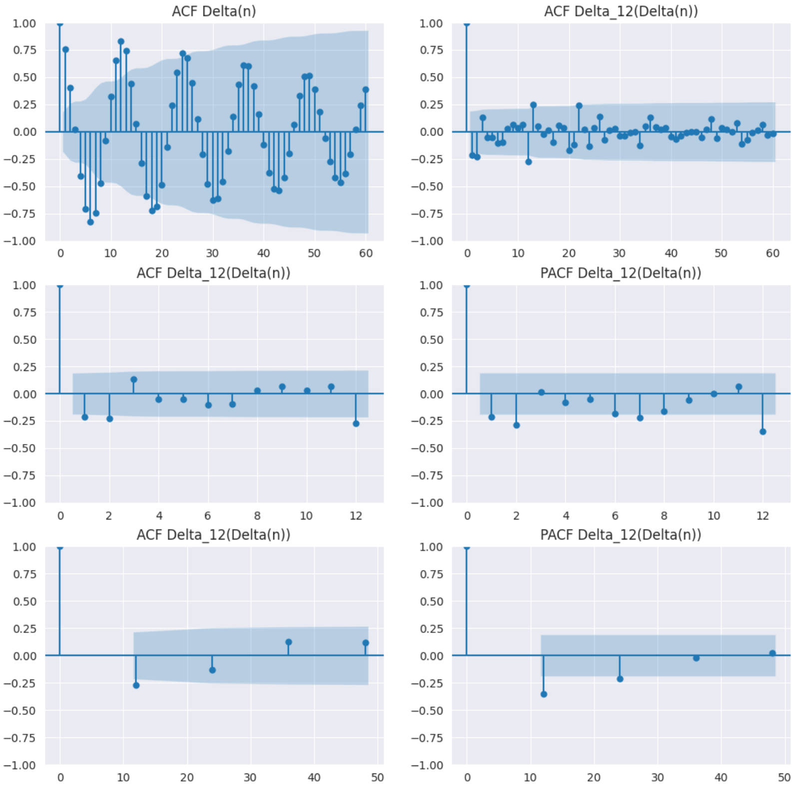


Figure 11: SARIMA ACF and PACF

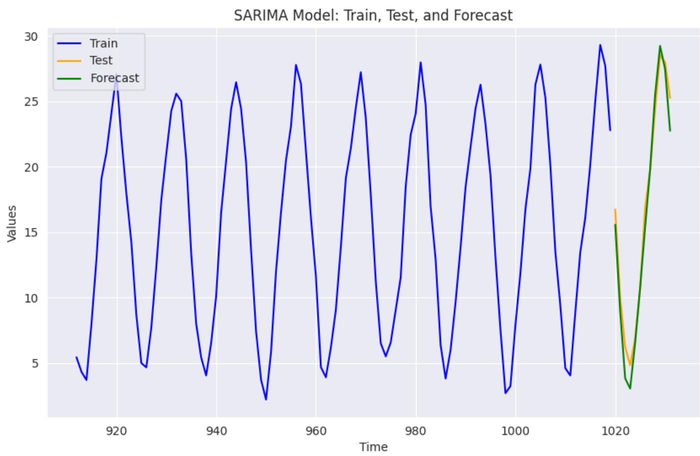


Figure 12: Sarima((5, 1, 3), (2, 1, 1), 12) forecast results

Given the extensive range of values tested, the grid search generated a significant number of models. For conciseness, we present only the results summarized in Table 3 and Figure 13, which highlight the performance of the best model identified: SARIMA((2, 3, 5), (1, 2, 1), 12). Full experiment details are annexed.

Interestingly, the initial model estimation based on ACF and PACF analysis outperformed the models generated by the grid search. This suggests that manual analysis of ACF and PACF

Measure	Value
p	[0 , 2]
d	[0 , 2]
q	[0 , 5]
P	[0 , 5]
D	[0 , 3]
Q	[0 , 3]
s	12

Table 2: Parameter ranges used for SARIMA grid search

Measure	Value
MSE	7.53
RMSE	2.74
MAE	2.40
MAPE	16.59

Table 3: Forecast performance metrics for SARIMA((2, 3, 5), (1, 2, 1), 12).

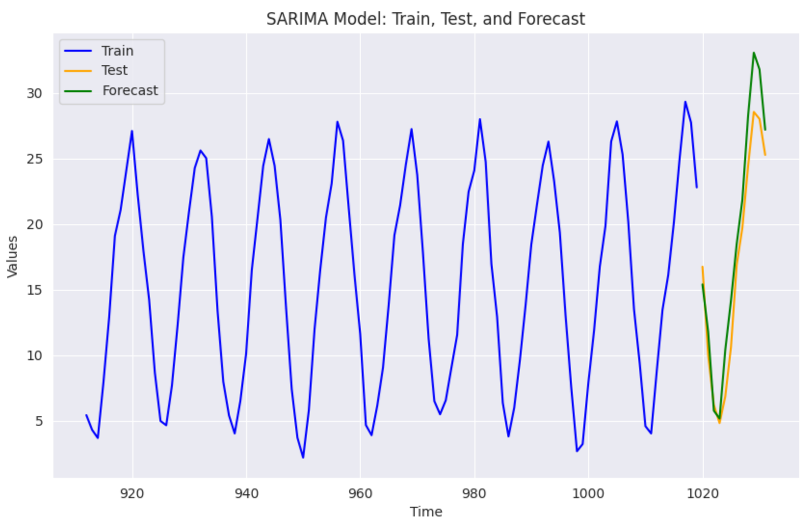


Figure 13: Forecast results for SARIMA((2, 3, 5), (1, 2, 1), 12).

values remains a valuable and effective approach to the selection of SARIMA models, despite the computational effort involved in the exhaustive search of the grid.

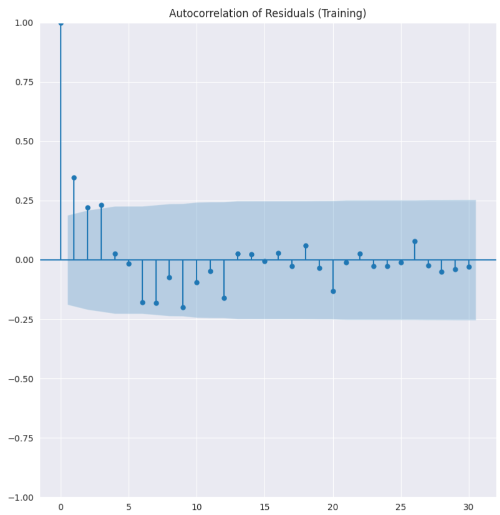
8.1.2. Residuals

To further assess the performance of the SARIMA models, we conducted an autocorrelation analysis of the training residuals to evaluate whether they exhibit any discernible patterns or significant autocorrelation. The two SARIMA models evaluated were considered.

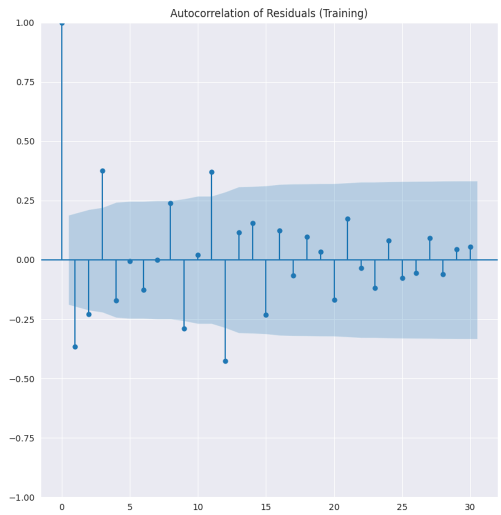
Figure 14 presents the autocorrelation plots of the residuals for both models. The key findings are as follows:

The residuals in the model from figure 14a exhibit minimal autocorrelation, with most lags falling within the confidence bounds, suggesting the model captures the temporal dependencies effectively and the residuals approximate white noise. On the other hand, some small spikes are observed in the first few lags, indicating some level of non-randomness in them and that the model can be better tuned to improve the results.

In contrast, the residuals from the grid search model 14b show more pronounced spikes at specific lags, indicating higher autocorrelation. These results suggest that the grid search model might not fully capture all temporal dependencies in the data, leading to patterns in the residuals.



(a) Residuals from SARIMA 1



(b) Residuals from SARIMA 3

Figure 14: Autocorrelation of the SARIMA models' Residuals

8.2. Exponential smoothing (ES)

Given the cyclical and seasonal nature of climate data, such as mean temperature, we determined that Triple Exponential Smoothing (TES) was the most suitable approach. Methods like SES and DES would be overly simplistic and fail to adequately capture the repeating seasonal patterns inherent in temperature data. TES accounts for these seasonal variations alongside trends, making it the ideal choice.

Figure 15 illustrates the training data, test data, and the TES forecast, along with the prediction intervals to highlight uncertainty:

As shown in the forecast plot, the predicted values closely align with the observed test data, validating the model's ability to represent seasonal temperature patterns.

8.2.1. Model Evaluation

We evaluated the model's forecast accuracy using various error metrics, which are summarized in table 4:

The relatively low error values suggest that the model performed well in capturing the seasonal and trend components of the time series. It is still noticeable that the best SARIMA model outperformed the TES approach.

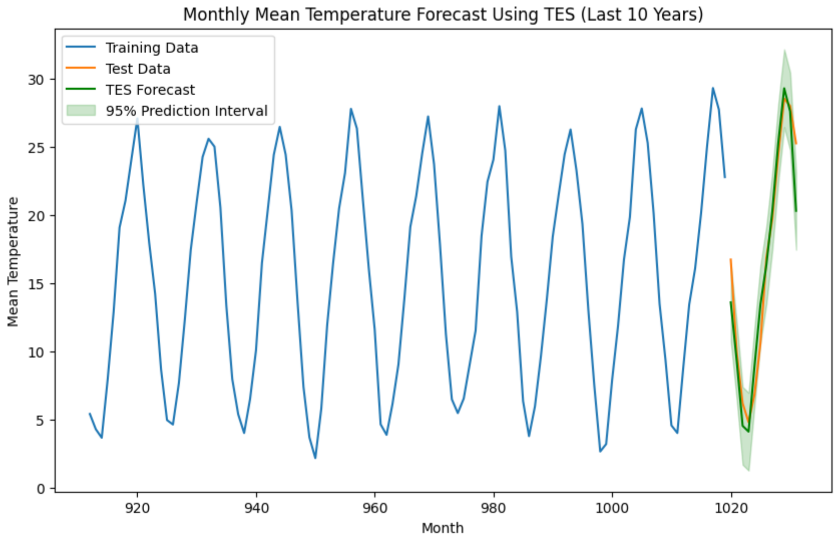


Figure 15: TES Forecast and Visualization

Measure	Value
MSE	4.36
RMSE	2.09
MAE	1.59
MAPE	13.01

Table 4: Forecast measures of TES

9. Time Series Selection Result

The cross-correlation results highlighted a few key variables that showed strong relationships with the mean temperature series. These variables were selected based on their high cross-correlation scores, indicating their relevance for modeling and prediction. The variables selected were: Maximum Temperature, Minimum Temperature, and Apparent Mean Temperature. The results are shown in figure 20.

These relationships are expected, as temperature metrics such as the maximum and minimum values are direct measures of daily temperature extremes, while the apparent temperatures factor in conditions like humidity that influence how temperature is experienced by humans.

10. Multivariate modelling and Forecasting Results

10.1. VAR

To analyze the interdependencies between mean temperature and other selected weather-related variables, we employed a Vector Autoregressive (VAR) model.

Figure 21 shows the forecasted (orange) and actual (green) values for each variable.

The model's accuracy was assessed using MAE, MSE, RMSE, and MAPE metrics, which can be checked in table 5.

In general, the metrics reveal that while the VAR model can capture some trends and patterns,

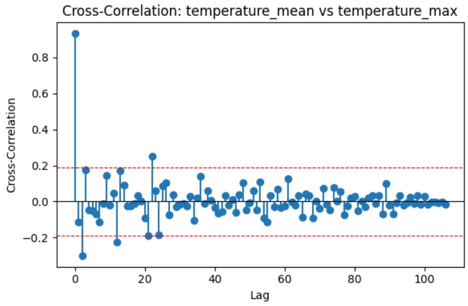


Figure 16: Temperature Max

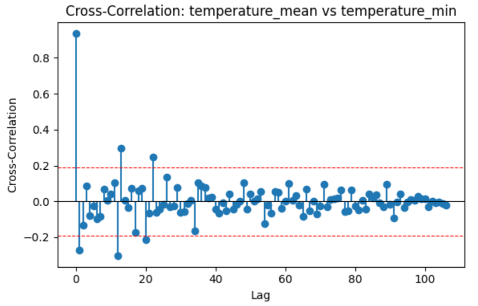


Figure 17: Temperature Min

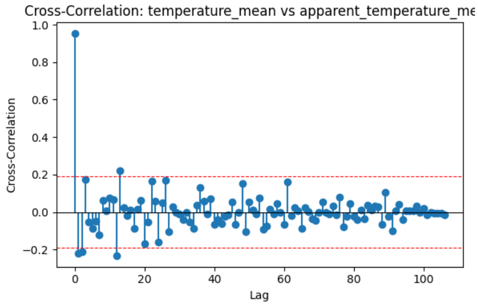


Figure 18: Apparent Temperature Mean

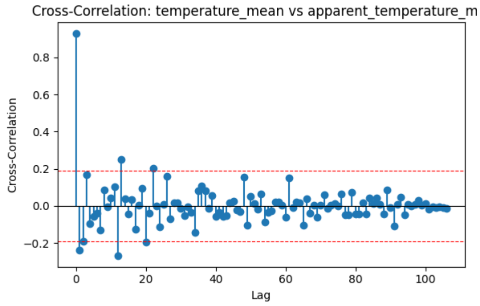


Figure 19: Apparent Temperature Min

Figure 20: Cross-correlogram results

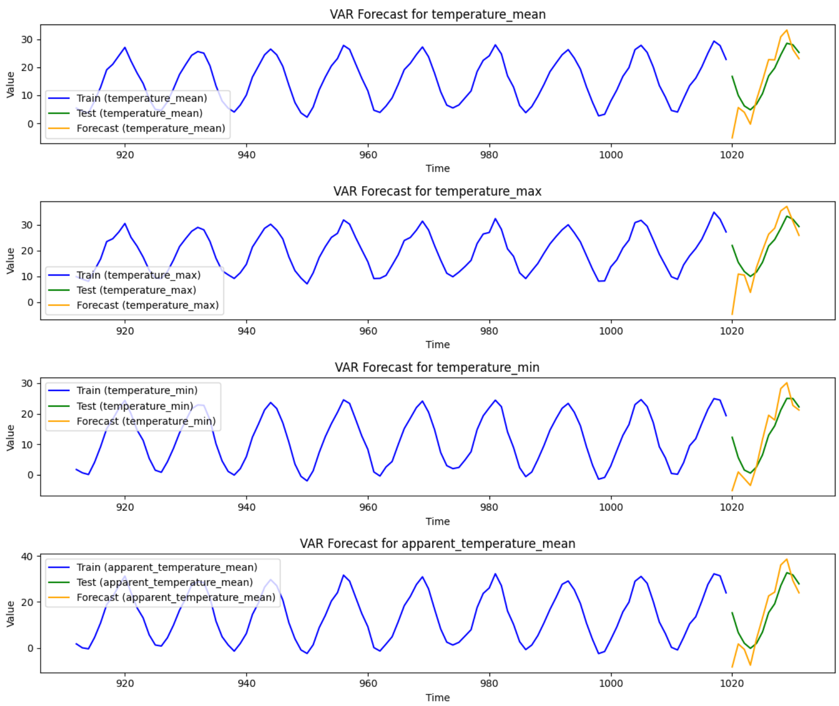


Figure 21: VAR Forecasts for all variables used

its linear structure and assumptions limit its capacity to accurately model the complex and non-linear relationships inherent in weather data. The high MAPE values for certain variables, particularly Minimum Temperature and Apparent Temperature Mean, further underscore the limitations of VAR in forecasting variables with high variability or extreme values. Improvements on this include exploring more sophisticated models.

Measure	MSE	RMSE	MAE	MAPE
Temperature Mean	55.72	7.46	5.30	40.79
Temperature Max	75.82	8.70	5.74	29.88
Temperature Min	42.17	6.49	4.85	117.59
Apparent Temperature Mean	74.28	6.66	1.41	686.32

Table 5: Forecast measures of VAR

Measure	MSE	RMSE	MAE	MAPE
Temperature Mean	0.0154	0.124	0.090	nan
Temperature Max	23.49	4.85	4.81	nan
Temperature Min	15.96	3.99	3.96	nan
Apparent Temperature Mean	9.87	3.14	2.84	nan

Table 6: SARIMAX forecast measures for every variable used

10.2. SARIMAX

The results obtained from the SARIMAX forecasting model for each variable are summarized in Table 6. The same hyperparameters used in the SARIMA model were applied to SARIMAX, allowing for consistency in assessing the impact of incorporating exogenous variables. However, MAPE values were not calculable (NaN) for all variables due to potential divisions by zero or missing data in the computation, as these low values led to undefined results.

Figure 22 presents the forecasted values for all the variables alongside their respective test sets. The forecasts and test lines appear to overlap significantly, indicating high accuracy for all variables, particularly for Temperature Mean, as suggested by its low MSE, RMSE, and MAE values.

For other variables, such as Temperature Max, the errors are higher, which may indicate limitations in the SARIMAX model's ability to capture more volatile patterns or relationships involving exogenous variables. These discrepancies could also arise from noise in the data or insufficient tuning of model parameters.

If we look at the residuals' autocorrelation plots also presented in figure 22 we can notice that the residuals exhibit almost no autocorrelation, with all lags falling within the confidence bounds. This indicates that the level of randomness in the residuals is really high, suggesting that this model is really efficient at capturing all temporal dependencies in the data.

Overall, the SARIMAX model demonstrates good performance for some variables while leaving room for improvement in others. Further tuning of hyperparameters and a deeper analysis of potential correlations between variables could enhance the model's predictive capabilities.

10.3. Multilayer Perceptron Model

For each month, we created a window of 12 months as input features, *X_train*. The target label for each window, *y_train*, corresponds to the temperature of the month immediately following the 12-month period. This process generated sequences of 12 months as inputs and their corresponding next month's temperature as the output.

Before proceeding with the forecast of future values, we initially performed a forecast on random samples from the test set to evaluate the model's performance. The results, as shown in table 7, demonstrated promising accuracy in predicting the patterns and trends of the time

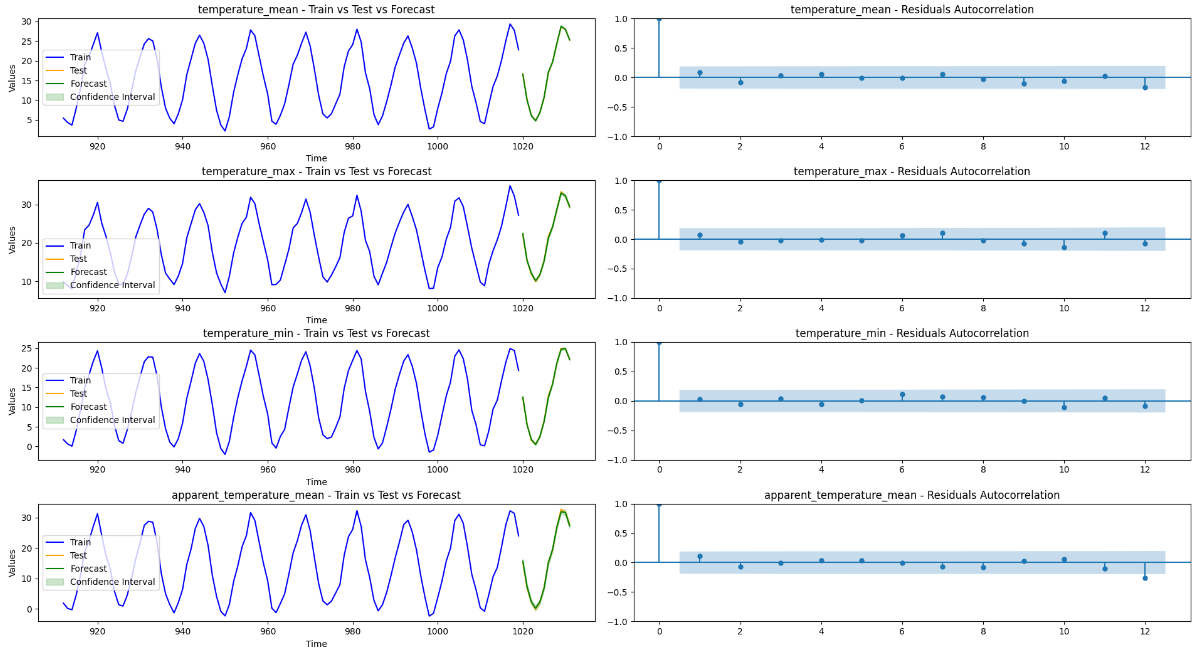


Figure 22: SARIMAX Forecasts for all variables used

series. In figure 23 we get indications that the result fits very well to the input test. Encouraged by these outcomes, we proceeded with the task of forecasting future values, specifically focusing on predicting the last year of the time series using the model trained on the earlier data.

Measure	Value
MSE	0.742
RMSE	0.861
MAE	0.700
MAPE	3.76

Table 7: Forecast measures of random input for MLP

The last year of the dataset was reserved for testing. This data was normalized using the same scale used for the training data, ensuring that the model processes the test data in the same way as the training data.

Once the model was trained, we used the trained model to generate forecasts for the next 12 months. To do this, we fed the model with the last 12 months of data as input. After predicting the next month's temperature, we appended the prediction to the input sequence and shifted the window forward by one month, generating predictions sequentially. This process continued for 12 months, producing a forecast for the entire test period.

Measure	Value
MSE	1.5172
RMSE	1.2318
MAE	1.0514
MAPE	9.03

Table 8: Forecast measures of last year input for MLP

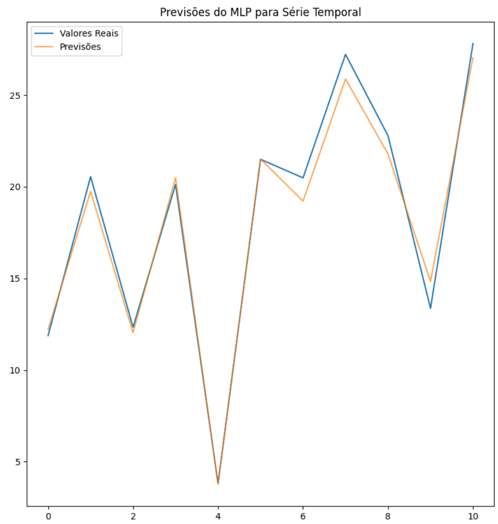


Figure 23: MPL Forecast with random input

As observed in the results presented in Table 8, there is a decline in performance compared to the training phase. However, Figure 24 illustrates that the model generally fits the real data well, with the exception of the minimum temperature values. This discrepancy could be attributed to the introduction of errors in the input data, as the model's predictions for future values are influenced by errors from previously predicted values as well.

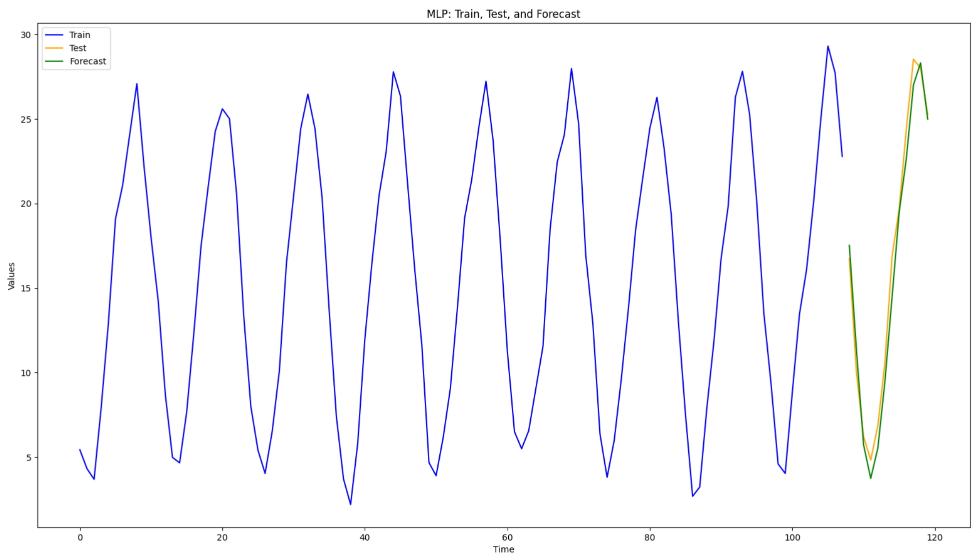


Figure 24: MPL Forecast with last year input

11. Discussion & Conclusion

This project has demonstrated the potential of time series analysis and forecasting methods to model and predict weather patterns effectively. By employing a diverse set of techniques, including decomposition, stationarity testing, and multivariate forecasting, we gained a comprehensive understanding of the dataset's characteristics, such as trends, seasonality, and noise components.

Among the tested models, SARIMA and TES proved to be the most effective for univariate forecasting, with SARIMA slightly outperforming TES in terms of accuracy metrics. For multivariate analysis, SARIMAX outperformed VAR by a wide margin, and provided valuable insights into inter-variable dependencies, although improvements in hyperparameter optimization could enhance predictive reliability. The MLP approach showed promising results for capturing nonlinear dynamics, albeit with some sensitivity to input errors.

Overall, the project highlights the importance of selecting appropriate models based on data characteristics and forecasting requirements. The integration of both traditional statistical techniques and machine learning models ensures robust and adaptable solutions for weather forecasting. Future work could focus on incorporating additional variables, enhancing model scalability, and exploring hybrid models to further improve forecasting performance.