Time Series Analysis on Solar Energy Consumption in United States

Allegra Chen,Weiting Lin

[a] *University of California Davis*

Abstract

In this project, our aim is to forecast solar energy consumption in the United States for the year 2023. The primary methodology employed for forecasting is time series analysis. However, additional data processing skills are also required to make the forecast more robust. Initially, we undertake exploratory data analysis and data processing. Subsequently, we apply the Moving Average method for detrending and deseasonalizing monthly solar consumption data. Based on these results, we fit the most appropriate ARMA model to the training data, capturing a wide range of autocorrelation structures. Ultimately, we will present the forecasted results of solar energy consumption for 2030 in the United States. The details of our methodology will be discussed in the following paragraphs.

1. Introduction

Nowadays, people are paying more attention to the development of solar energy because it plays an important role in reducing greenhouse gas emissions and mitigating climate change. Besides, it also offers additional benefits such as improving air quality and reducing water usage in energy production. This year, the UN Climate Change Conference in Dubai pledged on Dec. 2 to triple global renewable generation capacity by 2030. To be more specific, an analysis by S&P Global Commodity Insights mentions that the solar and wind capacity is forecasted to triple from the current levels.

However, Anna Mosby, the head of Environmental Policy Analytics at S&P Global, does not agree with this statement. She claims that the installed capacity of wind and solar energy will only double in the current base case. To determine the change in solar capacity by 2030, we assume that solar energy consumption has a positive relationship with solar energy capacity, because the supply market may align with the demand market. Therefore, solar energy consumption will be a good indicator to determine whether the solar energy capacity will increase or decrease to triple the current situation.

In this project, we will focus on solar energy consumption in the United States. To estimate solar energy consumption for 2023, we will employ time series analysis, using reasonable data processing methods and suitable ARMA (Autoregressive Moving-Average Model) models to predict the values for 2023.

2. Data description

We obtained data from the U.S. Energy Information Administration, which provides monthly data on solar energy consumption dating back to 1973. The data, available in a .csv file format, contains information on various sectors consuming solar energy. We aimed to analyze the total solar energy consumption; therefore, we only retrieved the columns for 'Solar Energy Consumption' and 'Month'. The time series plot of the data clearly shows an increasing trend in solar consumption over time, particularly in the last few years.

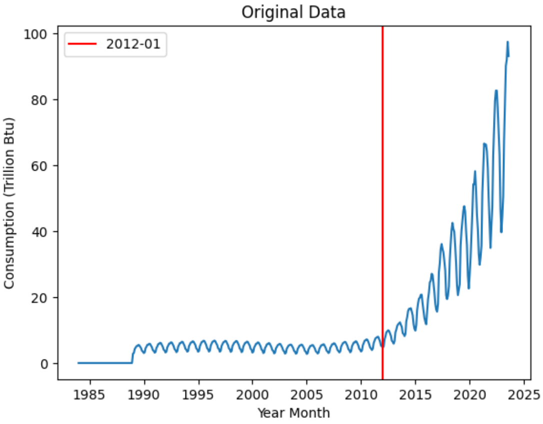

Figure 1: Solar energy consumption from January 1984 to August 2023

Statistically, we have 476 data points, with 22 of them representing unavailable values. The quantile data indicates that the third quartile is not far from the first, yet the maximum value of the data is 97.39, which is far from the third quartile. This phenomenon is reflected in the data plots, as the issue of solar energy has garnered more attention in recent years, leading to divergent values.

Statistics	count	Mean	std	min
	476	12.31	17.57	
	Q1	Q2	Q3	Max
	3.83	5.51	10.21	97.39

Table 1: Statistics information for solar energy consumption over year

3. Data analysis

3.1. Data processing

3.1.1. Data acquisition

Based on the time series plot of the data, Figure 1 above, we observed a clear increasing trend in solar energy consumption over the last few years. To enhance the precision of our 2023 solar energy consumption prediction, we aim to include data from January 2022. Additionally, the quantile data in the statistical information for solar energy consumption since 2012, Table 2, indicates that the third quartile is not far from the maximum value, similar to the quantile data presented in Table 1.

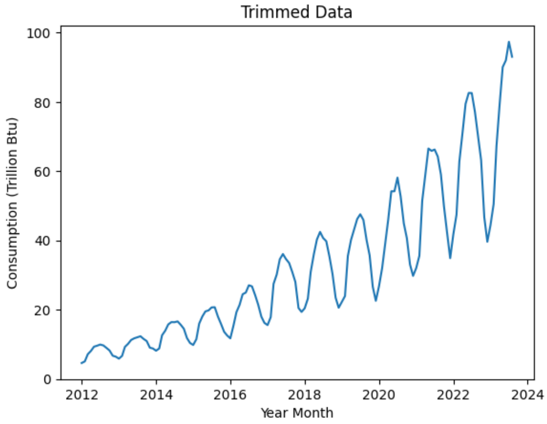

Figure 2: Solar energy consumption since January 2012

Statistics	count	Mean	std	min
	140	22.14	4.6	
	Q1	Q2	Q3	Max
	14.36	26.68	43.46	97.39

Table 2: Statistics information for solar energy consumption since 2012

Additionally, it is essential to divide the data into training and testing sets. In the forecasting section, we will use the testing data to assess the performance of our model. For this project, we have allocated 90% of the data for training and the remaining 10% for testing. The number of training data from January 2012 to June 2022 is 126, and the number of testing data from July 2022 to August 2023 is 14.

3.1.2. Data transformation

Before fitting ARMA models, it is essential to ensure that the data is stationary. However, the plot in Figure 2 suggests that the data's variance increases over time. Specifically, a greater distance between the peaks (high values) and valleys (low values) indicates higher variance, as it shows that the data points are more widely dispersed from the mean. We employed two methods to stabilize the variance in the data over time.

- Log transformation (logarithmic transformation) The Log transformation is defined as:

$$Y_t' = \log(Y_t), \text{ where } Y_t \text{ is a data point at time t}$$

In this method, we apply a logarithmic transformation to every data point in the training set. As evidenced by Figure 3, this appears to stabilize the variance, as the distance between peaks and valleys over time becomes more consistent.

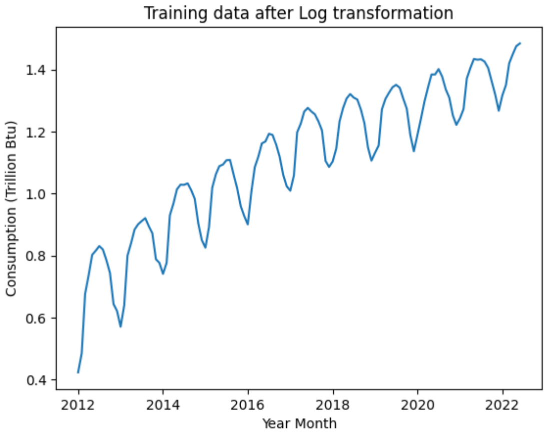

Figure 3: Log transformation method

- Box-Cox transformation

The Box-Cox transformation is defined as:

$$y(\lambda) = \begin{cases} \frac{Y_t^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0, \\ \ln(Y_t) & \text{if } \lambda = 0. \end{cases}, \text{ where } Y_t \text{ is a data point at time t}$$

We used the above formula to transform the training data, and the results shown in Figure 4 provide evidence that the transformed training data has stable variance over time.

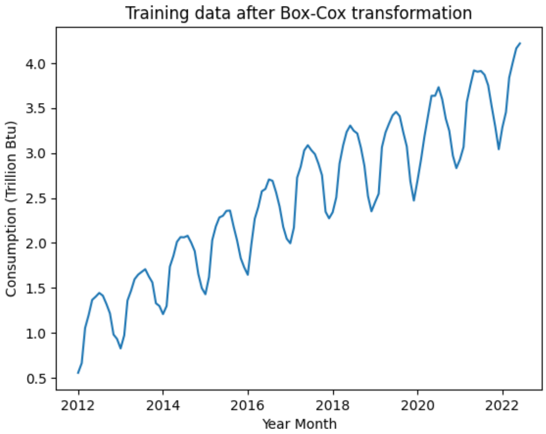

Figure 4: Box-Cox transformation method

To determine which method performed better in variance stabilization, we can observe the range of solar energy consumption values in the two methods. It appears that the Log transformation method does a better job of stabilizing variance, as the range of solar energy consumption in the Log transformation method is nearly 1.2, compared to 4 in the Box-Cox transformation method.

Hence, we decided to use the Log transformation method to transform the training data.

3.2. *Time series decomposition*

As illustrated by the time series plot after data transformation, there is a clear additive relationship between trend, seasonality and residual. Naturally, the additive model for time series decomposition is proposed as the following:

$$Y_t' = \log(Y_t) = m_t + s_t + X_t,$$

where $t \in \mathbb{Z}$, and the relationship between the time t and the corresponding year j and month k, $1 \le k \le 12$ can be formulated as $t = k + 12(j-1)$. Y_t is the solar energy consumption at time t decomposed by the trend component m_t, the seasonal component s_t, and the residual X_t.

3.2.1. *Trend & seasonality decomposition*

The very first step in forecasting with the proposed additive model is to estimate the trend and seasonal components so that they can be eliminated from the additive model to separate the residual component for further analysis. The moving average method is applied here to (1) detrend the data using a moving average filter, (2) estimate the seasonal components based on detrended data and (3) re-estimate the trend based on deseasonalized data.

(1) Detrending: Since the period is 12 (even), the following moving average formula with adjusted weights and a window of size 13 is used to estimate the trend at time t:

$$\hat{m}_t = \frac{1}{12}(.5y_{t-6}' + y_{t-5}' + \cdots + y_{t+5}' + .5y_{t+6}')$$

The estimated trend and detrended data is present in Figure 5 and 6

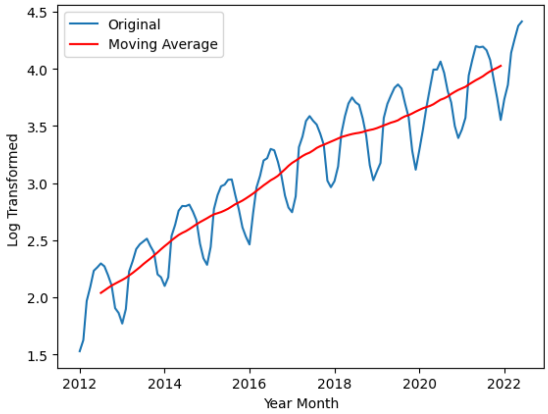

Figure 5: Trend estimate using the moving average filter applied on the training data

(2) Seasonality estimation: The detrended time series plot in Figure 6 illustrates very strong seasonal components, which is part of the additive model proposed previously. The seasonal components for month k can be computed as

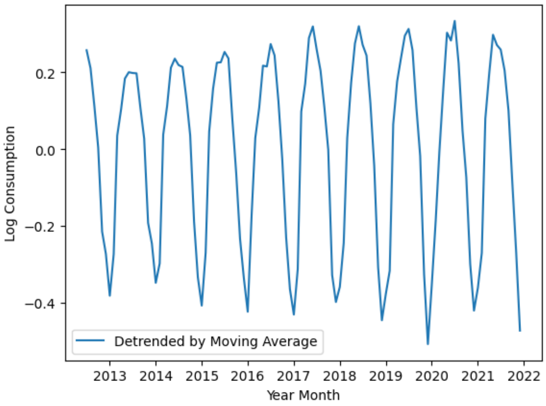

Figure 6: Detrended training data

$$\hat{s}_k = \mu_k - \frac{1}{12}\sum_{l=1}^{12}\mu_l, \quad k = 1, \cdots, 12$$

where

$$\mu_k = \frac{1}{N-1}\sum_{j=2}^{N}(y_{k+12(j-1)}' - \hat{m}_{k+12(j-1)}), \quad k = 1, \cdots, 6$$

$$\mu_k = \frac{1}{N-1}\sum_{j=1}^{N-1}(y_{k+12(j-1)}' - \hat{m}_{k+12(j-1)}), \quad k = 7, \cdots, 12$$

and $\hat{s}_k = \hat{s}_{k-12}$ whenever $k > 12$. The seasonal component extracted from the detrended data is illustrated in Figure 7.

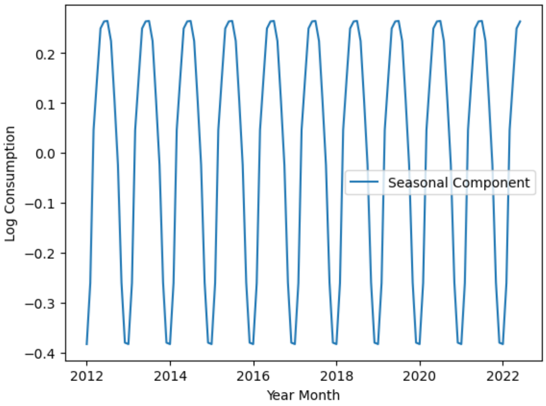

Figure 7: Seasonal components illustrated at different time over the years.

(3) Trend re-estimation: To re-estimate trend, a polynomial regression of degree d would be a natural choice observing there is a clear almost linear trend in Figure 8. Separating out the seasonality estimated in the last step from the log-transformed training data, now trend can be re-estimated properly on the deseasonalized data as:

$$\hat{m}_t = y_t' - \hat{s}_t = \hat{\beta}_0 + \hat{\beta}_1 t + \cdots + \hat{\beta}_d t^d$$

After doing a stepwise regression, the degree $d = 4$ is selected based on its relatively low AIC value of -392.1 and high adjusted $R^2 = 0.994$. Though the coefficients for the degree 3 and 4 terms are tiny, they are still preserved considering the log scale of data in general. The final fitted model is

$$\hat{m}_t = 1.9102 + 0.0184t + 0.0002t^2 - 3.89\times10^{-6}t^3 + 1.75\times10{-}8t^4$$

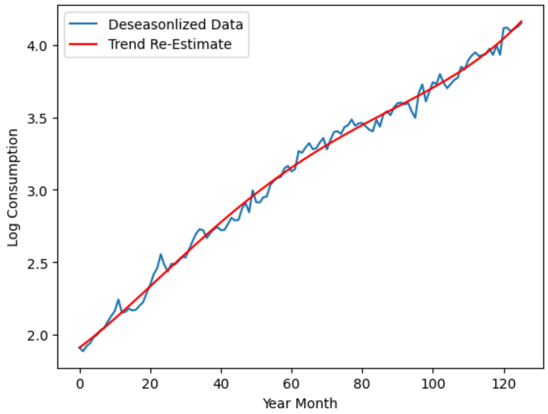

Figure 8: Trend re-estimated with a polynomial regression of degree 4 (red curve) on the deseasonlized data (blue curve)

3.2.2. *Stationary test on residuals*

In time series analysis, ensuring data stationarity is essential. We employed the ADF test (Augmented Dickey-Fuller test) to ascertain the presence of trends within the residuals. Additionally, we plotted the mean and variance of residuals grouped by year to determine if the residuals were stationary.

Based on Figure 9, we notice that the variance and mean of residuals grouped by year are sufficiently close. Additionally, the ADF test results presented in Table 3 indicate that there is no significant trend in the residuals.

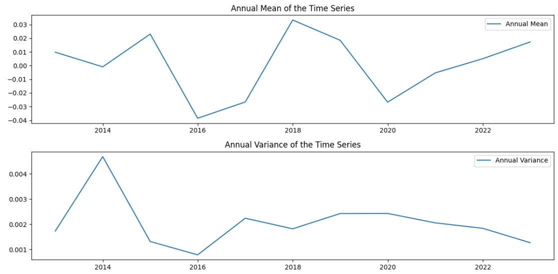

Figure 9: variance and mean of residuals grouped by year

ADF Statistic	p-value	result
-6.264	4.146e-08	no trend, p-value> 0.05

Table 3: ADF test

With the residuals plot in Figure 10, it is evident that the residuals are evenly distributed around y-axis at 0. Hence, we concluded that the residuals are stationary and suitable for fitting time series models.

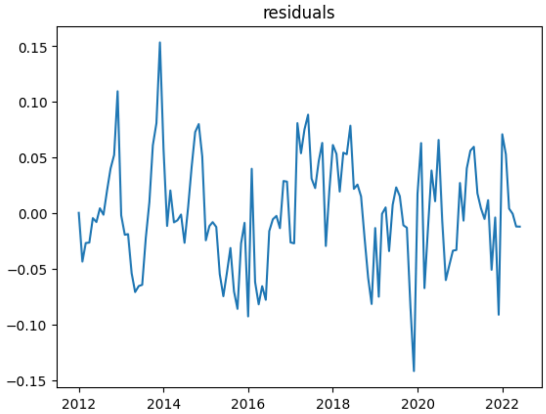

Figure 10: Residuals plot after detrending and deseasonalizing the transformed data

Besides, the Ljung-Box test indicates that the residuals exhibit significant autocorrelation because the p-values are smaller than 0.05 in every lag. Hence, we can employ time series models to capture this autocorrelation in stationary time series data.

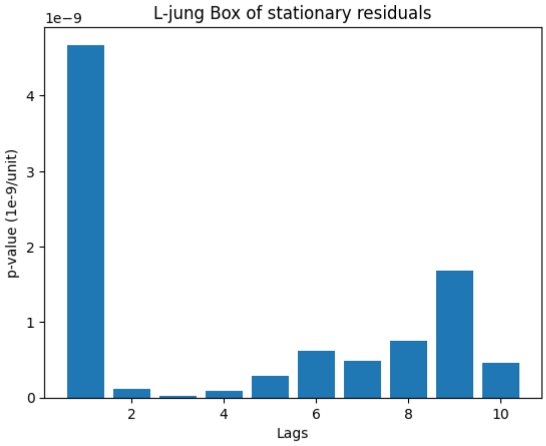

Figure 11: Ljung-Box plot of residuals

3.3. *Data modeling*

After achieving stationary residuals data in the previous process, we can fit a time series model, such as ARMA (Autoregressive Moving-Average), AR (Autoregressive), or MA (Moving Average), to our data. Before fitting a model to achieve optimal forecasting performance in time series analysis, examining the ACF (Autocorrelation) and PACF (Partial Autocorrelation) plots is essential. These plots help determine the order (number of lags) we will use in the time series model.

3.3.1. *ACF and PACF plots analysis*

In the ACF plot (Figure 16), the autocorrelation values decrease or 'tail off' over the lags. This pattern suggests the need for an ARMA or AR model to capture the autocorrelation structure of the time series data effectively.

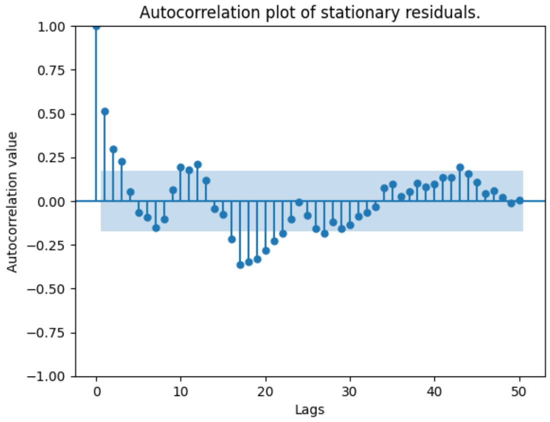

Figure 12: Autocorrelation plot after detrending and deseasonalizing the transformed data

To determine the number of lags for an AR or ARMA model, we need to examine the PACF plot (Figure 13). In this plot, the trend appears to cut off at lags 1, 9, 10, and 14. Therefore, we initially fitted an AR model using these specific lag values, where the partial autocorrelation value at each lag exceeds 95% of the Bartlett confidence interval.

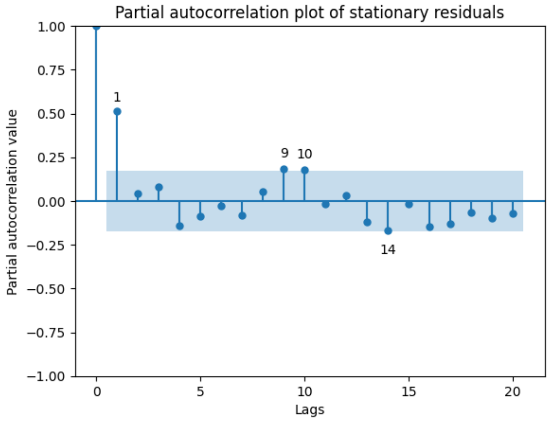

Figure 13: Partial autocorrelation plot of stationary residuals

3.3.2. *Model fitting*

- AR(Autoregressive) model We first fitted AR models to the residuals data at lags 1, 9, 10, and 14, separately. Subsequently, we selected the best AR model based on the smallest AIC (Akaike Information Criterion) value. This approach favors the model that best explains the data using the fewest parameters, thereby achieving a balance between goodness of fit and simplicity.

  With AIC plot in Figure 14, the AR(1) model indicated the smallest AIC value close to -435, so we selected AR(1) as the best model.

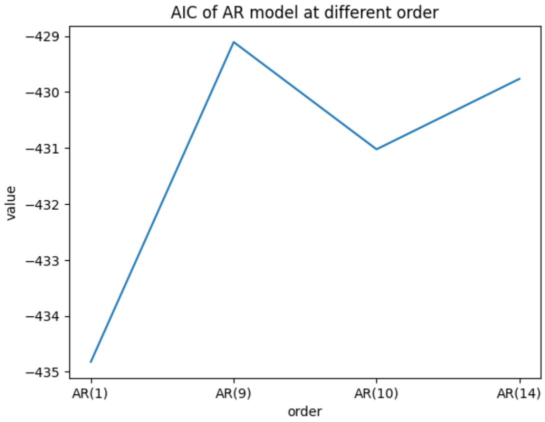

Figure 14: AIC value of different AR models

To verify whether the AR(1) model efficiently captures the autocorrelation among the data, the Ljung-Box test was used to determine the result. Based on Figure 15, it is evident that the p-values are greater than 0.05 at each lag, leading us to conclude that the model meets our expectations

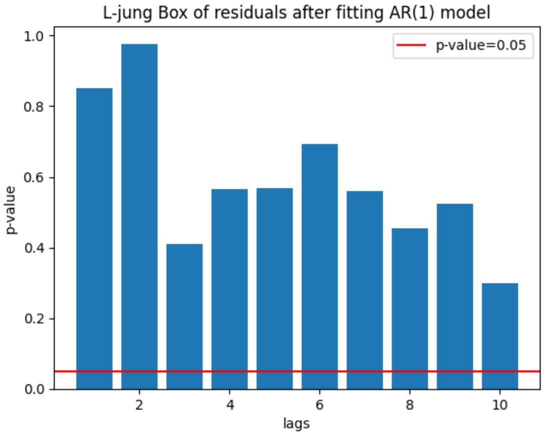

Figure 15: Ljung-Box plot of residuals from AR(1) model

We are also interested in the ACF plot of residuals from the AR(1) model, and according to Figure 16, there is still a significant lag at 17. Therefore, we conclude that a Moving Average (MA) component might need to be considered. To address this issue, we will try fitting an ARMA model to fully capture the autocorrelation structure of the residuals

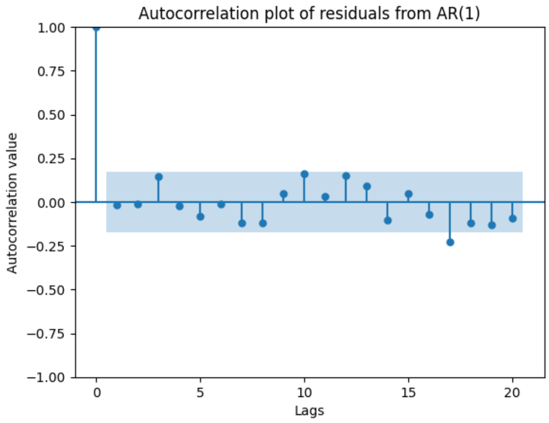

Figure 16: ACF plot of residuals from AR(1) model

- ARMA(Autoregressive moving-average model) model

  By using a for loop to iterate through MA and AR orders from 1 to 5, we obtained 25 AIC values for various ARMA models. We applied the same method to determine the best ARMA model using the AIC. Ultimately, the best-performing model was identified as ARMA(2,3) and AIC value is -436.40 in Table 4

	1	2	3	4	5
1	-433.09	-431.25	-431.78	-430.28	-428.34
2	-431.12	-429.97	-436.40	-434.97	-431.32
3	-430.59	-429.78	-434.66	-433.68	-430.66
4	-431.24	-432.23	-434.09	-431.77	-431.62
5	-429.10	-427.44	-425.27	-424.07	-430.33

Table 4: AIC of ARMA(p,q), p: row index, q: column index

In the ACF plot shown in Figure 17, it is evident that there are no significant lags. To reinforce this observation, we used the Ljung-Box test to verify the result

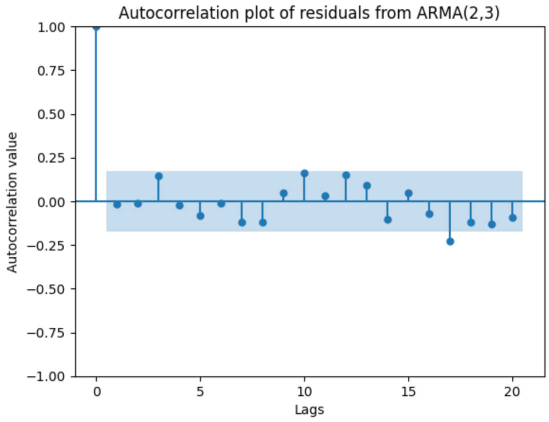

Figure 17: ACF plot of residuals from ARMA(2,3) model

Further analysis using the Ljung-Box test provides evidence that the ARMA(2,3) model has successfully captured the time series' autocorrelation structure.

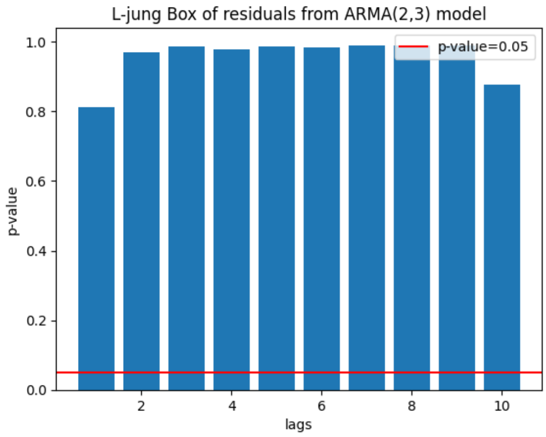

Figure 18: Ljung-Box plot of residuals from ARMA(2,3) model

The residuals plot from the ARMA(2,3) model resembles white noise, indicating that the model is well-fitted.

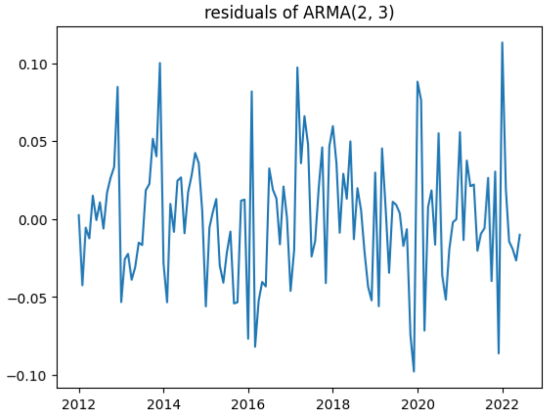

Figure 19: residuals plot of ARMA(2,3) model

3.3.3. Model selection

After analyzing various models, we ultimately chose the ARMA(2,3) model due to its smallest AIC and ability to fully capture the autocorrelation structure of the data. In the following section, we forecast the training data using ARMA(2,3) and discuss its forecasting performance.

3.4. Forecasting

In this section, we will discuss the forecasting process, which plays a crucial role in determining forecasting performance. Additionally, we have employed multiple indicators to quantify the accuracy of the forecast.

Moreover, there are 2 questions we are interested in answering by forecasting. First, how well does the proposed model perform in forecasting? This is addressed by forecasting the consumption of the testing data and comparing the forecast consumption and the actual observed consumption in the testing data. Second, will the solar energy consumption triple in 2030?

3.4.1. Forecasting process

The forecasting process can be further broken down into the following 4 steps: *(1) Residual forecasting*: Using the fitted ARMA(2, 3) model, the residuals $\hat{X}_t$'s of the testing data can be forecasted. *(2) Trend forecasting*: Using the fitted polynomial regression of degree 4 model, the trend $\hat{m}_t$ for future time could be forecasted. *(3) Seasonality forecasting*: Since seasonality repeats every period of 12 months, they are already estimated by $\hat{s}_k$ and no further prediction is needed. *(4) Consumption forecasting*: Adding together the forecasted residual, trend and seasonality, it provides the forecasted log consumption, $\hat{Y}'_t$. By taking exponetial over $\hat{Y}'_t$, the forecasted consumption can be computed at any time t, illustrated in Figure 21.

3.4.2. Forecasting performance

Following the process described above, we forecasted solar energy consumption from July 2022 to August 2023, the period covered in our testing data. We used several indicators to quantify the performance.

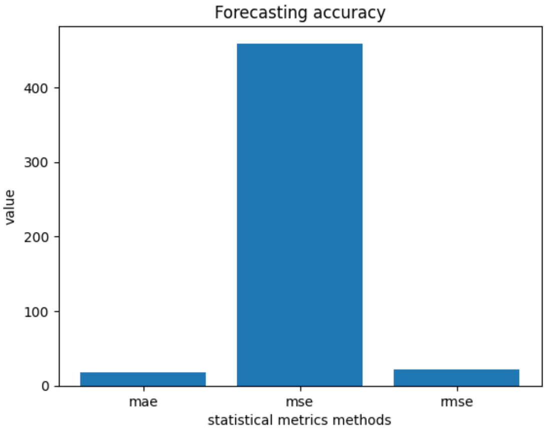

Figure 20: Examination of forecasting performance

We also use the R^2(R-squared) method to assess forecasting performance. With a value of 0.9019, we conclude that our forecast is approximately 90% accurate.

In Figure 21, it is observed that the forecasts of the testing data closely align with the actual data values starting from July 2022.

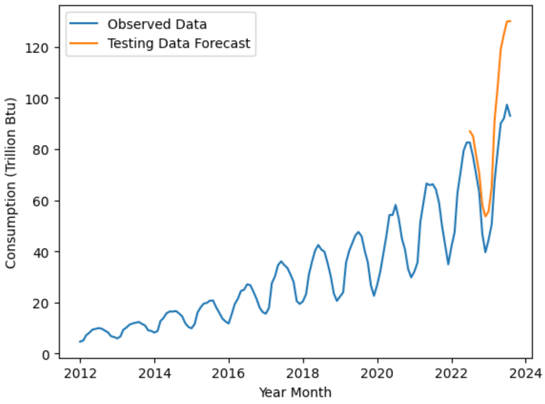

Figure 21: Consumption forecasting on the testing data (yellow curve) compared with the actual data (blue curve)

3.4.3. Forecasting result

Following the process above, the forecast of solar consumption in January 2030 is 2292431.18 trillion Btu whereas the solar consumption in January 2023 was 55.58 trillion Btu. Solar energy consumption is expected to triple by 2030, or possibly even more.

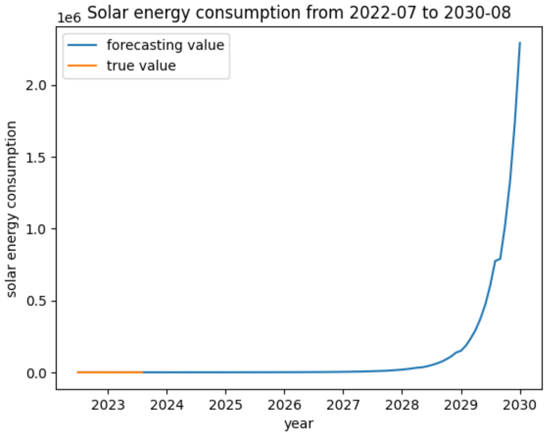

Figure 22: Consumption forecasting from July 2022 to August 2023)

4. Discussion

In Figure 10, a slight variance difference is observed before and after 2017. Examining the trimmed data in the time series plot of Figure 2, it is noticeable that solar energy consumption from 2012 to 2017 was not as high as the consumption after 2017. This might explain why different variances over the years persist even after detrending and deseasonalizing.

To address this issue, we could start by selecting data from after 2017 and then proceed with detrending and deseasonalizing to see if the variance difference over time still exists. We believe that under this approach, the residuals from the detrending and deseasonalizing processes would be more stationary.

In addition, an interesting phenomenon worth mentioning is that after detrending the original data in seasonality decomposition, the seasonality in June (at the peak of each season) appears to increase from 2013 to 2017 where as the seasonality in January (at the trough of each season) appears to decrease from 2013 to 2017. This could induces fitting a regression for the peak and the trough months. This could also help in making the residuals more stable comparing to the approach in the paper which the seasonality is treated fixed for each month over all years.

5. Conclusion

Based on our forecasting results for January 2030, we believe that solar energy consumption will rapidly increase in the coming decades. Governments worldwide need to be cautious about solar energy prediction policies and ascertain if any underlying crises have been overlooked in light of the rising solar energy consumption. According to an article in the Nature Journal, it is mentioned that solar production is highly related to land usage. This could pose a challenge and bring additional issues to the world, such as environmental hazards and impacts on terrestrial carbon stocks.

6. Reference

- COP28: Leaders pledge to triple renewable generation capacity by 2030
- U.S. Energy Information Administration (EIA)
- Solar Energy Consumption Data
- Time Series From Scratch—Train/Test Splits and Evaluation Metrics
- The potential land requirements and related land use change emissions of solar energy
- Solar Energy, Wildlife, and the Environment
- Notes from Professor Alexander Aue

7. Appendix

- STA137 Final Project Code .py file
- STA137 Final Project Code Colab file