Time Series Analysis Assignment 3:

A Time Series Analysis and Forecast of Total Energy Load

Authored by: Phuong Pham, Sothearaot Tat

Professor Name: Dr Haydar Demirhan

15 June 2025

1. Introduction

In recent years, forecasting in the energy markets has emerged as a high-impact application of machine learning and statistical modelling, particularly in the context of transitioning toward sustainable and renewable energy systems. Specifically, the dataset is a time series observation of five major Spanish cities' electrical infrastructure, consisting of their consumption, generation, pricing, and weather conditions.

The goal of this study is to enhance operational decision-making within the energy industry by accurately predicting the national electrical load. According to the respective forecasts, system operators can ensure the alignment between electricity generation and prospective market demands. This balance helps crucially prevent power outages due to underproduction while avoiding the economic and environmental inefficiencies of overproduction. The predictive capabilities are also believed to achieve significant cost optimization, improve energy efficiency, and ensure more stable grid operation.

In addition, the objective refers to conducting a comprehensive time series analysis of the aggregated electrical load. Even though the data points were observed hourly, this project considers the weekly load to fit the proper models that are capable of forecasting the next 10 units ahead of time. Besides ARIMA, which is the most appropriate for time series analysis, SARIMA is also taken into account due to potential seasonality. Model specification will be guided by formal modeling techniques, including analysis of autocorrelation structures (ACF and PACF), seasonal decomposition, and model selection criteria. The finalized model will be evaluated through residual diagnostics and used to generate a comprehensive forecast.

2. Methodology and Limitation

Methodology

The dataset captures the hourly electrical records over a four-year period, from the beginning of 2015 to the end of 2018, resulting in a total of over 35 thousand observations and almost 30 variables. For time series modeling purposes, the hourly data points were initially transformed using a weekly grouping approach, as the focus of this analysis is shifted to analyzing the weekly average electricity consumption.

Thus, the final time series object comprises weekly observations from week 1 of 2015, with a frequency of 53 weeks every accounting year, respectively accommodating for the ISO-standard year. In particular, the timestamp feature is converted to appropriate European date-time formats before splitting into each week and its corresponding year identifiers. The hourly consumption will then be aggregated to the weekly average load for further analysis. In fact, although the informal calendar year only has 52 weeks, which equals 364 days, the internationally recognized full year accounts for 365 or 366 days (a leap year such as 2016), leading to a 53rd week that appears to take the leftover days beyond the 52 full weeks.

After storing the result as a time series object with a frequency of 53, the initial diagnostics begin with the visualization of the time series to assess the five descriptive characteristics . The Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) are plotted to reveal the presence of any potential seasonal components with the specified periodicity, consistent with annual cycles in energy demand. On top of that, the stationarity assumption is confirmed using the Augmented Dickey-Fuller (ADF), Phillips-Peron (PP), and Kwiatkowski-Phillips-Schmidt-Shin (KPSS) tests.

Following the seasonal structure, the one-order seasonal differencing is used alongside 53 lags to remove the yearly seasonal component. The resulting residual plots are re-checked using seasonal ACF and PACF to identify remaining autocorrelation patterns before fitting models to validate whether the seasonal differencing alone could stabilize the series, or additional non-seasonal terms are required. ACF, PACF, and Extended ACF (EACF) are performed, accompanied by the generated BIC table for model order selection purposes. In detail, the combinations of AR and MA terms are chosen based on the peaks and cut-offs in ACF/PACF, significant entries in EACF matrices, and the lower values from BIC subset selection. The prospective candidates' SARIMA are then tested under CSS, ML, and CSS-ML methods, followed by the final evaluation using AIC/BIC criteria via a custom function sort.score(); residual diagnostic plots of histogram, qq plot, and relative seasonality and normality tests; and the estimates of forecast accuracy metrics such as ME, RMSE, MAE, MPE, MAPE, MASE, and ACF1. The best-performing models are considered those with the lowest corresponding scores, well-behaved residuals, and marginal error measures.

Limitation

While the structured framework for time series analysis and forecasting is well-defined, it has several inherent limitations. Though the linearity and stationarity assumption of the SARIMA models are satisfied using transforming and differencing techniques, the presence of structural changes or non-linear dynamics common to make the model sensitive to outliers and lack a mechanism to account for unexpected shocks or exogenous variables, especially in this energy market. Furthermore, while weekly aggregation reduces noise, it also removes finer temporal patterns that could be valuable for high-resolution forecasting. Overfitting or underfitting may be a potential issue due to the partially subjective model selection process guided by those statistical indicators. In this situation, the importance of careful model validation and the potential advantages of combining with other data-driven approaches in further evaluations are substantially weighted.

3. Findings and Discussion

3.1. Time Series Overview

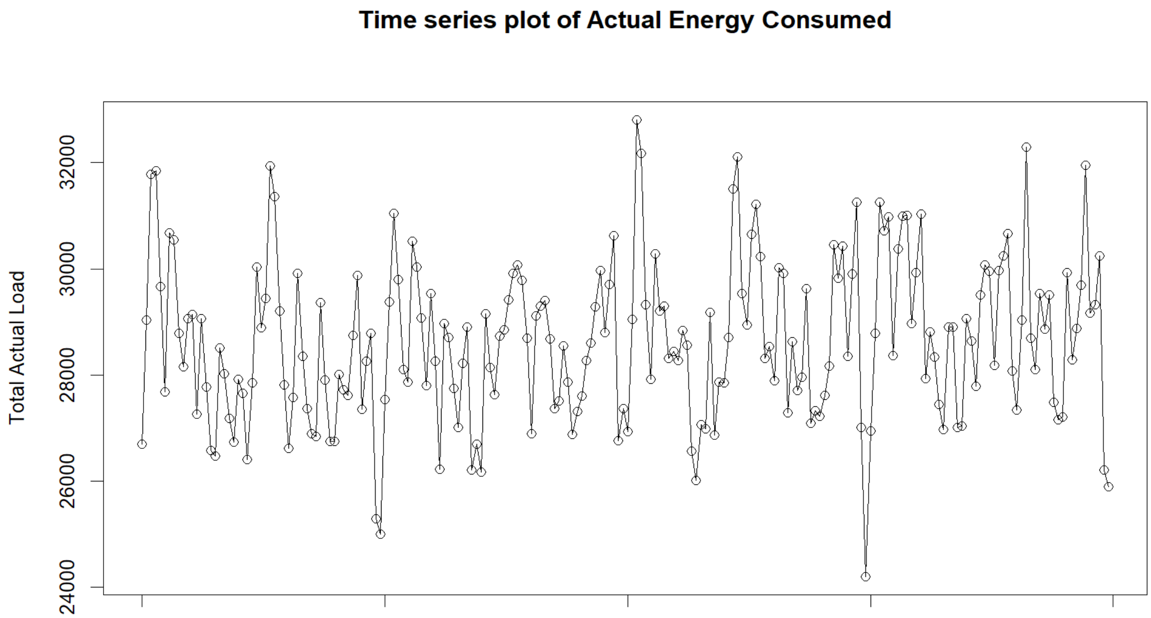


Figure 1: Time Series Plot of Weekly Energy load

Figure 1 is the time series plot demonstrating the highlighted seasonal effects and relative patterns in average energy load over time. There are five key attributes of the series that must be focused on: “trend”, “seasonality”, “changing variance”, “behavior”, and “change point”. The consumption is witnessed to follow a clear seasonal pattern, with consistent spikes and dips roughly every year. Looking closely, these peaks are likely associated with summer and winter months, when heating/cooling demand increases due to significant changes in temperature. Besides, troughs are seen during milder seasons, especially experiencing sudden drops at the end of every year. Since the load fluctuates significantly and repeats its pattern every 53 weeks, the plot tends to lack a clear trend that indicates stationarity to justify the use of the SARIMA model with seasonal differencing rather than non-seasonal differencing. In terms of changing variance, the variance appears to be fairly constant regardless of fluctuations in amplitude. As a result, no critical structural change points are captured in this time series visualization. Isolated sharp drops and peaks resulting in an autoregressive (AR) behavior. According to the stability of this energy series with no obvious pattern of either increasing or decreasing variance, variance-stabilizing transformations like Box-Cox or Logistic may not be strictly necessary. To verify the assumption, Appendix B displays the comparison between the original series and the post-Box-Cox-transformed series. With an optimal lambda = -0.3 applied, the output visualization shows no significant difference compared to the original object, confirming the insufficiency of the transformation in this circumstance.

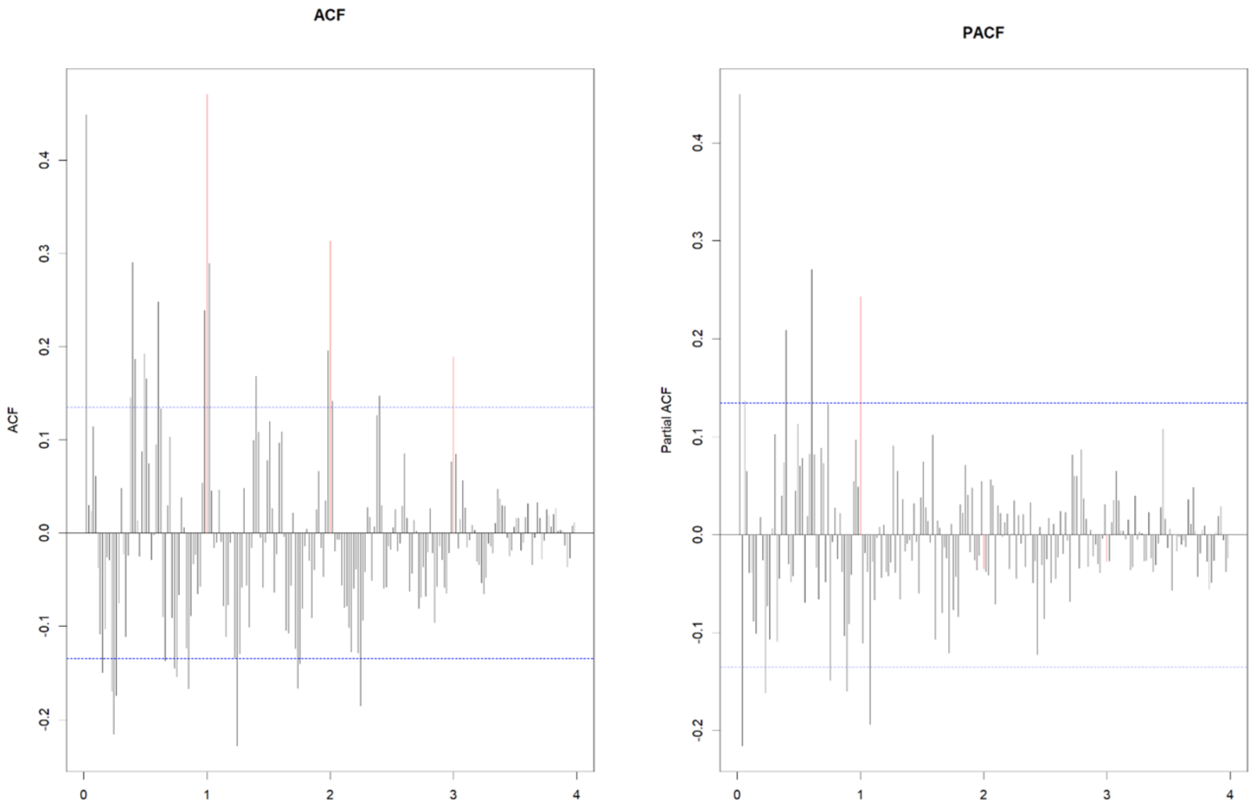


Figure 2: ACF and PACF of weekly time series data of energy load

Figure 2 illustrates the autocorrelation function and partial autocorrelation function plots of the weekly time series data of total energy load. The pair of plots exhibits patterns characteristic of non-stationary seasonal trend. The ACF shows slowly decaying pattern at seasonal lags, while PACF shows one high autocorrelation at the first lag, with subsequent lags having no significant autocorrelation. These features are highly indicative of seasonal non-stationarity in the data. Next, the first seasonal-differenced series will be considered.

3.2. Seasonal Differencing

The first seasonal differencing (see: Figure 3) resulted in the loss of a season of data, in other words, 53 data points, which are shown as 0 in the residual plot in the first season. In general, no significant variations are observed between plots of the original series and the seasonal-differenced series. Specifically, the differenced series still maintain clear seasonality and show no evidence of long-term trend.

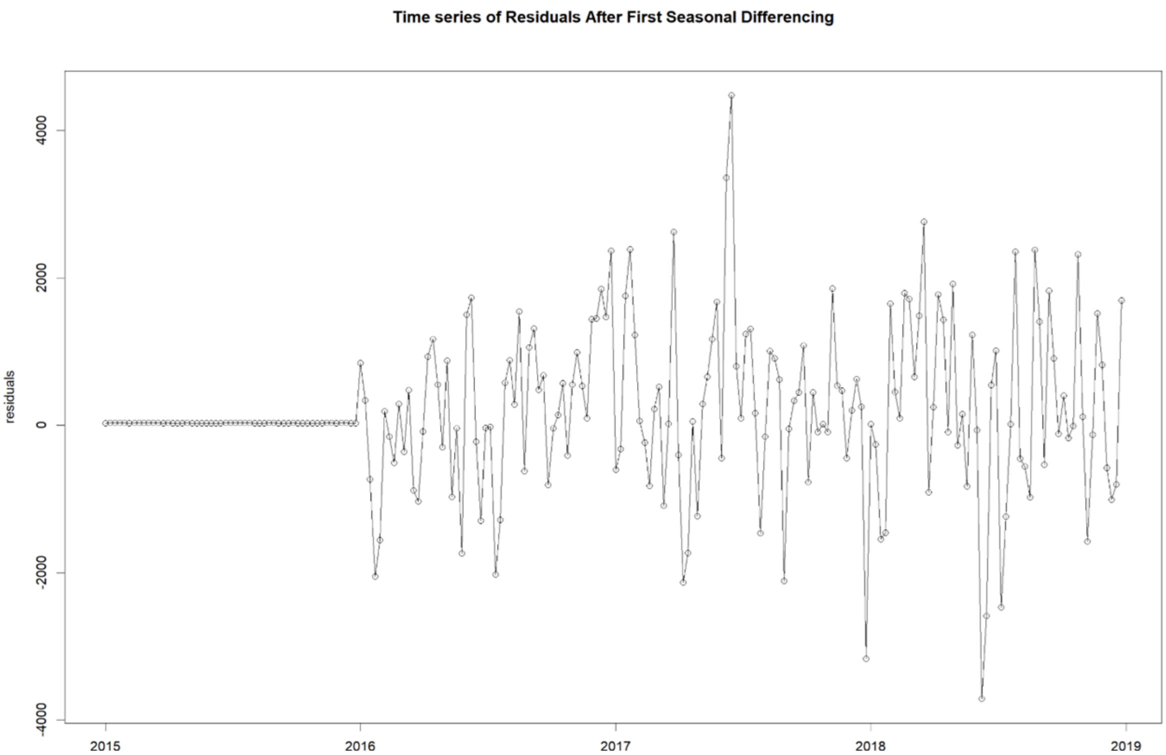
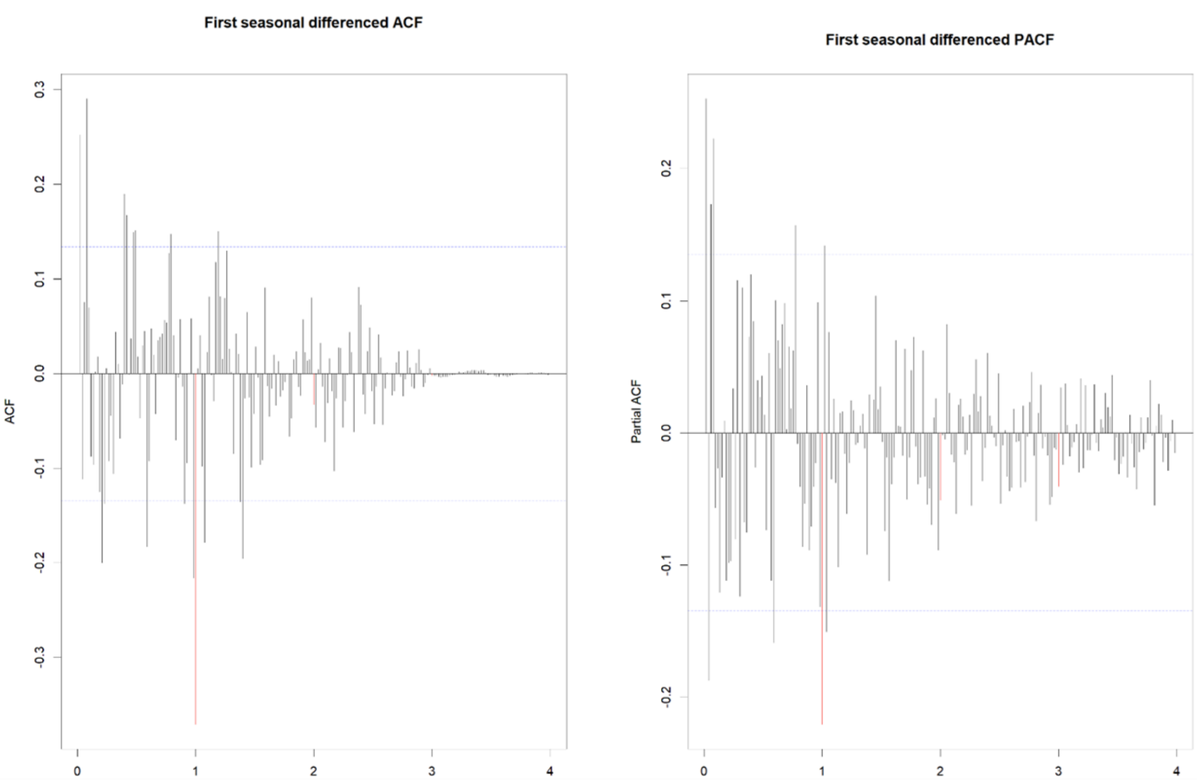


Figure 3: Residual time series plot after first seasonal differencing



n

Figure 4: ACF and PACF of first-order seasonal differenced series

the data is addressed. The ACF has only 1 significant autocorrelation at the first seasonal lag after which there is a clear cut-off as opposed to the slowly decaying pattern seen in the previous ACF plot. This implies one significant order of seasonal moving-average component (SARIMA(p,d,q)(p,1,1$)_{53}$). In addition, the PACF now appears to exhibit a slowly decaying pattern although the seasonal autocorrelation after the first seasonal lag were insignificant. This leads to uncertainty as to whether there is a non-significant seasonal autoregressive component or one order of significant seasonal AR. Due to this uncertainty, potential models will be fitted using two seasonal components specification, SARIMA(p,d,q)(0,1,1$)_{53}$ and SARIMA(p,d,q)(1,1,1$)_{53}$. The results from the models fitting shall be reviewed to ascertain whether setting the seasonal AR to 1 is beneficial to the analysis.

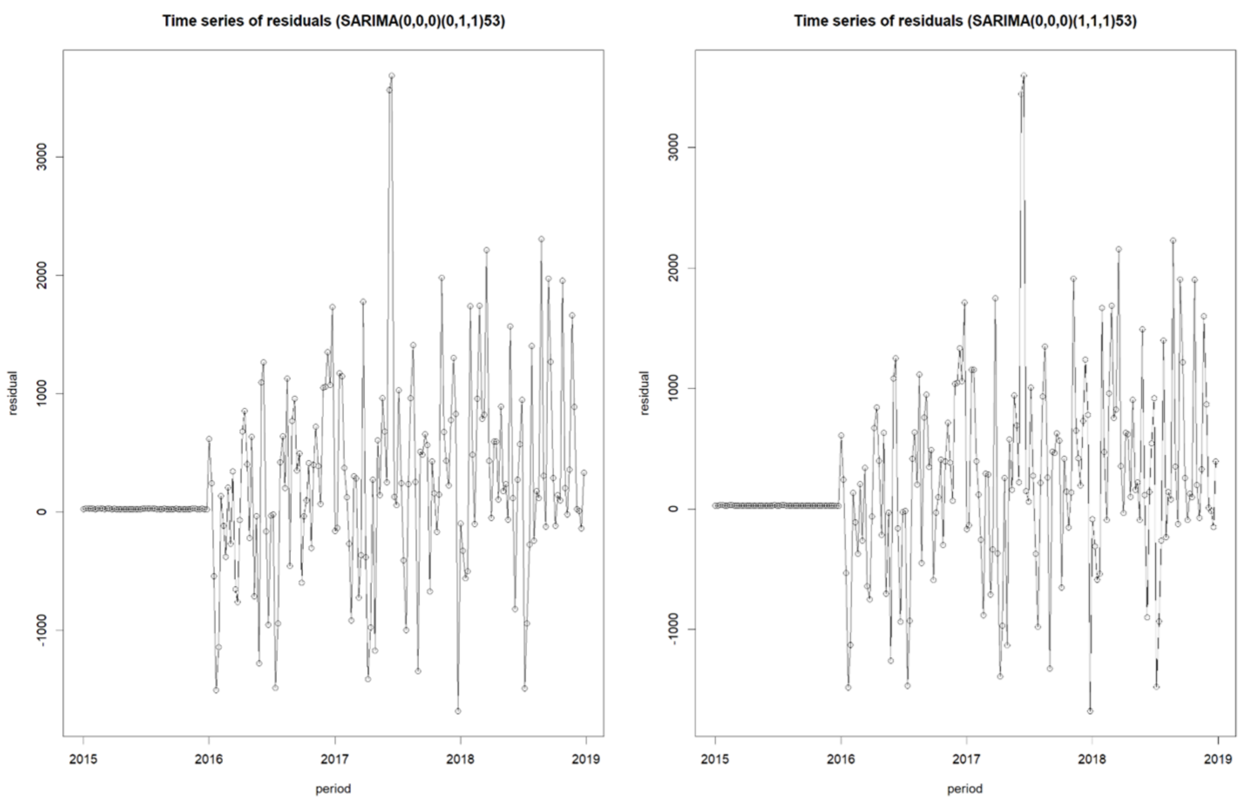


Figure 5: Time series plots of residuals after applying first seasonal differencing with seasonal AR 1 and 0

Figure 5 compares the time series of the residuals after implementing the first seasonal differencing with the one order of seasonal AR and MA, to the time series of residuals after differencing with one order of seasonal MA without any seasonal AR components. Overall, the two series appear to be identical, displaying no significant discrepancies. Therefore, it is expected that the results of the various model specification tools will not vary much between the two residual time series. Additionally, these plots are similar to the time series after only applying the seasonal differencing *(figure 3)*. One key difference is that the value around mid 2017, which also looked to be a potential outlier in the time series after only the first differencing (seasonal ar and ma = 0), now appears to stand out even clearer in residuals plots after incorporating other seasonal components. This suggests an outlier in the differenced series which may cause possible issue of non-normal residuals distribution after models fitting.

3.3. Model specification

Before specifying non-seasonal components of the models, statistical tests are performed to check whether the residual after the first seasonal differencing remains stationary (see: *figure 6*).

```
	Augmented Dickey-Fuller Test

data:  ts
Dickey-Fuller = -5.7164, Lag order = 5, p-value = 0.01
alternative hypothesis: stationary


	Phillips-Perron Unit Root Test

data:  ts
Dickey-Fuller Z(alpha) = -154.67, Truncation lag parameter = 4, p-value = 0.01
alternative hypothesis: stationary


	KPSS Test for Level Stationarity

data:  ts
KPSS Level = 0.84171, Truncation lag parameter = 4, p-value = 0.01
```

Figure 6: Result of ADF, PP, and KPSS test on residuals after first seasonal differencing

The ADF and PP test statistics were –5.7164 and –154.67, respectively (btoh p-value < 0.05). There is statistically significant evidence to suggest that the series is stationary. However, the KPSS test shows test statistic 0.84171, p-value < 0.05, which suggests that the series is non-stationary. This conflicting finding suggests that although the series has no unit roots, thus is likely stationary, it may have a deterministic trend which indicates that the series is trend stationary, although not difference stationary. As such, it is not necessary to implement further differencing to the residual series.

Figure 7 shows the respective pairs of ACF and PACF plots for the residuals of both the model with first order of seasonal differencing, AR and MA, and the model with only the first order of seasonal differencing and MA (seasonal AR = 0). Again, the two models have similar residual outcomes. The seasonal autocorrelation in all ACF and PACF plots are insignificant, indicating that adding one order of seasonal autoregressive component would not be beneficial and could instead cause the models to be overly

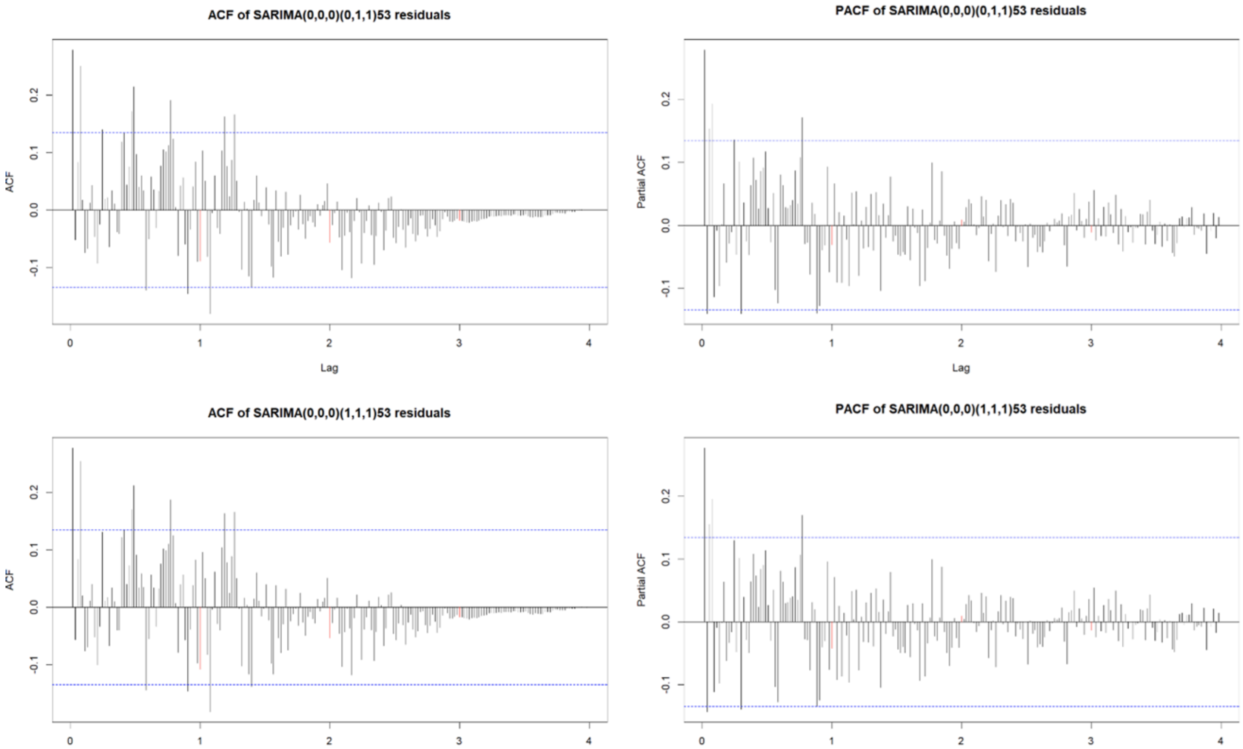


Figure 7: ACF and PACF residuals after applying first seasonal differencing with seasonal AR 1 (top) or 0 (bottom)

complex, which could lead to model instability or overfitting. As such, this report will focus on the models specified with only one order of seasonal differencing and MA.

From the ACF and PACF plot, considering only significant correlation before the first seasonal lag (as it is expected that non-seasonal AR and MA behaviours repeat each season), there are 8 significant autocorrelations in both ACF and PACF. This gives SARIMA(8,0,8)(0,1,1)$_{53}$. On the other hand, this model incorporates one near-miss autocorrelation in each of the ACF and PACF plots. Disregarding this near-miss autocorrelation, another model can be specified, SARIMA(7,0,7)(0,1,1)$_{53}$.

AR/MA	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	x	o	o	x	o	o	o	o	o	o	o	o	o	o
1	x	o	o	x	o	o	o	o	o	o	o	o	x	o
2	x	o	o	x	x	o	o	o	o	o	o	o	x	o
3	x	x	x	x	o	o	o	o	o	o	o	o	o	o
4	x	x	x	x	o	o	o	x	o	o	o	o	o	o
5	o	x	x	x	o	o	o	o	o	o	o	o	o	o
6	o	x	x	x	x	x	o	o	o	o	o	o	o	o
7	o	x	x	o	x	x	o	o	o	o	o	o	o	o

Figure 8: Extended correlation function of the residual time series after seasonal differencing and MA = 1

The extended autocorrelation function (EACF) is also used to determine candidate models to forecast the weekly total energy load (see: *figure 8*). A clear corner of o's can be seen extending from the intersection of AR = 0 and MA = 4. Selecting the surrounding o's to include as possible non-seasonal AR and MA components, the candidate models found using the EACF are SARIMA(0,0,4)(0,1,1)$_{53,}$ SARIMA(1,0,4)(0,1,1)$_{53,}$ SARIMA(0,0,5)(0,1,1)$_{53,}$ SARIMA(1,0,5)(0,1,1)$_{53}$.

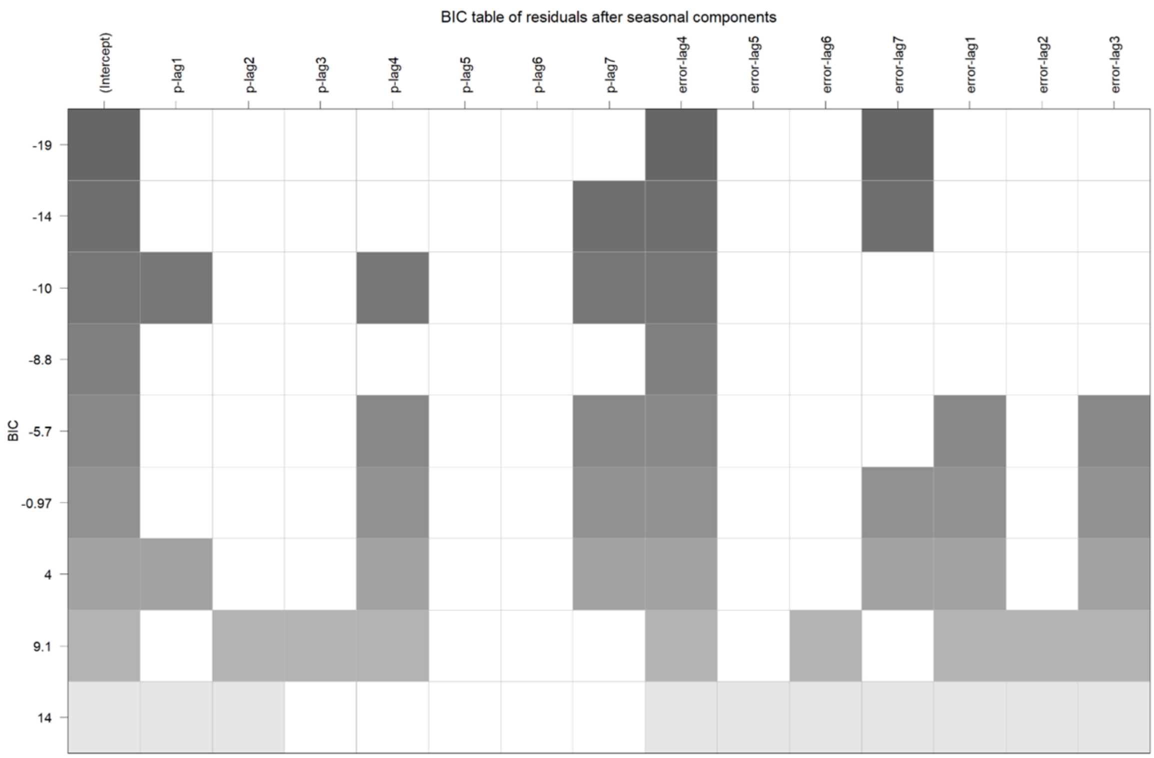


Figure 9: BIC table of the residuals after applying SARIMA(0,0,0)(0,1,1)$_{53}$

The final model specification tool used to determine candidate models was the Bayesian Information Crierion (BIC), particular the BIC table. The best model suggested by the table is one that is associated with the lowest BIC score. These models were SARIMA(0,0,4)(0,1,1)$_{53}$ and SARIMA(0,0,7)(0,1,1)$_{53.}$ Particularly, models with 4 orders of non-seasonal MA and 7 order of non-seasonal AR seem to be well supported by the BIC table. Ranking below the two models mention earlier is SARIMA(7,0,4)(0,1,1)$_{53}$. Below this were SARIMA(4,0,4)(0,1,1)$_{53}$ and SARIMA(1,0,4)(0,1,1)$_{53.}$ These models are selected as possible models for the analysis.

Overall, the pool of candidate models include SARIMA(8,0,8)(0,1,1)$_{53}$, SARIMA(7,0,7)(0,1,1)$_{53}$, SARIMA(0,0,4)(0,1,1)$_{53}$, SARIMA(1,0,4)(0,1,1)$_{53}$, SARIMA(0,0,5)(0,1,1)$_{53}$, SARIMA(1,0,5)(0,1,1)$_{53}$, SARIMA(0,0,7)(0,1,1)$_{53}$, SARIMA(4,0,4)(0,1,1)$_{53}$, SARIMA(7,0,4)(0,1,1)$_{53}$.

3.2. Model Fitting

9 SARIMA models were fitted to the time series data of weekly aggregated total energy load using 3 methods – CSS, ML, CSS-ML. A test of coefficients was conducted for each of the models (see: ***Appendix A***). In all models, the first order seasonal moving average coefficient (SMA(1)) was significant at confidence level of 0.05. The first model, SARIMA(8,0,8)(0,1,1)$_{53}$, was unreliable when fitted using CSS and ML as it could not estimate standard errors for some of the coefficients. Using CSS to determine the initial values then fitting the model using ML (CSS-ML) allowed for standard errors to be computed. However, only a few of the 17 coefficients (8 MA terms, 8 AR terms and 1 seasonal term) were significant, implying that this model may be overfitting and mis-specified. Higher order models such as SARIMA(7,0,7)(0,1,1)$_{53}$ and SARIMA(0,0,7)(0,1,1)$_{53}$ also had many insignificant coefficients, which indicate they may not be optimal for the current objective. The SARIMA(7,0,4)(0,1,1)$_{53}$, on the other hand, when estimated using CSS yielded statistically significant estiamtes for all coefficients except for the 7$^{\text{th-}}$ order autoregressive coefficient. Conversely, using CSS or CSS-ML, this model produced many fewer significant coefficients, casting doubts on whether this model is effective in analysing the time series. The SARIMA(4,0,4)(0,1,1)$_{53}$ had only 1 significant coefficient, which was the SMA(1) term. Similarly, SARIMA(1,0,5)(0,1,1)$_{53}$ produced only two significant coefficients, which were SMA(1) and MA(4) when using CSS. Using ML or CSS-ML, however, reliable estimates for this model could not be made. Only 3 models were consistent across all methods. These were SARIMA(0,0,5)(0,1,1)$_{53}$, SARIMA(1,0,4)(0,1,1)$_{53}$, and SARIMA(0,0,4)(0,1,1)$_{53}$. In addition to SMA(1), the MA(4) term was invariably significant in all 3 models. A significant MA(1) coefficient was also common in these models except for SARIMA(1,0,4)(0,1,1)$_{53}$, in which this moving average term was not significant. From the tests of the coefficients, SARIMA(0,0,4)(0,1,1)$_{53}$ appears to be the most promising model.

3.3. Model Selection

Models	df	AIC
SARIMA(1,0,5)(0,1,1)53_ml	8	2659.553
SARIMA(1,0,5)(0,1,1)53_CSSML	8	2659.603
SARIMA(0,0,4)(0,1,1)53_ml	6	2661.304
SARIMA(0,0,4)(0,1,1)53_CSSML	6	2661.304
SARIMA(1,0,4)(0,1,1)53_CSSML	7	2662.275
SARIMA(1,0,4)(0,1,1)53_ml	7	2662.275
SARIMA(0,0,5)(0,1,1)53_CSSML	7	2662.351
SARIMA(0,0,5)(0,1,1)53_ml	7	2662.351
SARIMA(7,0,7)(0,1,1)53_CSSML	16	2664.967
SARIMA(7,0,4)(0,1,1)53_CSSML	13	2665.108
SARIMA(8,0,8)(0,1,1)53_ml	18	2665.959
SARIMA(0,0,7)(0,1,1)53_CSSML	9	2666.312
SARIMA(0,0,7)(0,1,1)53_ml	9	2666.312
SARIMA(7,0,7)(0,1,1)53_ml	16	2667.15
SARIMA(4,0,4)(0,1,1)53_CSSML	10	2667.294
SARIMA(4,0,4)(0,1,1)53_ml	10	2667.294
SARIMA(7,0,4)(0,1,1)53_ml	13	2668.608
SARIMA(8,0,8)(0,1,1)53_CSSML	18	2671.735

Models	df	BIC
SARIMA(0,0,4)(0,1,1)53_ml	6	2679.717
SARIMA(0,0,4)(0,1,1)53_CSSML	6	2679.717
SARIMA(1,0,4)(0,1,1)53_CSSML	7	2683.757
SARIMA(1,0,4)(0,1,1)53_ml	7	2683.757
SARIMA(0,0,5)(0,1,1)53_CSSML	7	2683.834
SARIMA(0,0,5)(0,1,1)53_ml	7	2683.834
SARIMA(1,0,5)(0,1,1)53_ml	8	2684.105
SARIMA(1,0,5)(0,1,1)53_CSSML	8	2684.155
SARIMA(0,0,7)(0,1,1)53_CSSML	9	2693.932
SARIMA(0,0,7)(0,1,1)53_ml	9	2693.932
SARIMA(4,0,4)(0,1,1)53_CSSML	10	2697.983
SARIMA(4,0,4)(0,1,1)53_ml	10	2697.983
SARIMA(7,0,4)(0,1,1)53_CSSML	13	2705.004
SARIMA(7,0,4)(0,1,1)53_ml	13	2708.504
SARIMA(7,0,7)(0,1,1)53_CSSML	16	2714.07
SARIMA(7,0,7)(0,1,1)53_ml	16	2716.253
SARIMA(8,0,8)(0,1,1)53_ml	18	2721.2
SARIMA(8,0,8)(0,1,1)53_CSSML	18	2726.975

Figure 10: BIC and AIC scores of all models estimates using ML and CSS sorted in ascending order

After fitting, the models estimated via the ML or CSS-ML method were evaluated using the BIC as well as AIC scores. The model with the lowest BIC or AIC score is generally considered optimal as it balances between the number of parameters and the model fit. Recall that SARIMA(1,0,5)(0,1,1)$_{53}$ had unreliable estimates for some of its coefficient. Therefore, although the model has the lowest AIC score, it should still be disregarded as this instability could lead to inaccurate forecast estimates. From the sorted AIC and BIC scores of the models above, the optimal model appears to be SARIMA(0,0,4)(0,1,1)$_{53}$.
However, this method of model selection excludes models estimated using the CSS method. To compare across all models, various error metrics are computed such as mean error (ME), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Percentage Error (MPE), Mean Absolute Percentage Error (MAPE), Mean Absolute Scaled Error (MASE), and Lag-1 Autocorrelation of the residuals (ACF1).
Figure 11 shows the respective error measures for each of the models using the three different methods. The table also includes comments on the model's reliability during fitting, as well as whether the model performs the best in terms of specific error metrics, i.e whether the model produce any minimums among the errors. The model which should have performed the best was SARIMA(8,0,8)(0,1,1)$_{53}$ but due

Models	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Comments
SARIMA(8,0,8)(0,1,1)53_ML	70.0509	648.8674	427.2288	0.2052	1.4787	0.4482	-0.0040	Unreliable
SARIMA(7,0,7)(0,1,1)53_ML	70.2019	649.4020	430.7933	0.2015	1.4917	0.4519	0.0072	Min(RMSE, MAE, MAPE, MASE)
SARIMA(0,0,4)(0,1,1)53_ML	137.2523	689.6864	457.4748	0.4349	1.5822	0.4799	-0.0112	Best BIC/AIC model
SARIMA(1,0,4)(0,1,1)53_ML	125.3614	688.0337	459.0234	0.3943	1.5878	0.4816	-0.0264	
SARIMA(0,0,5)(0,1,1)53_ML	129.4609	687.5638	459.0804	0.4085	1.5878	0.4816	-0.0270	
SARIMA(1,0,5)(0,1,1)53_ML	74.8354	711.3114	475.5217	0.2160	1.6468	0.4989	-0.0003	Unreliable
SARIMA(0,0,7)(0,1,1)53_ML	127.7980	687.5328	458.4163	0.4028	1.5857	0.4809	-0.0273	
SARIMA(4,0,4)(0,1,1)53_ML	117.8818	685.2266	456.8674	0.3681	1.5802	0.4793	-0.0237	
SARIMA(7,0,4)(0,1,1)53_ML	118.8796	684.6739	454.7931	0.3721	1.5700	0.4771	-0.0215	
SARIMA(8,0,8)(0,1,1)53_CSS	134.6727	762.0736	501.3503	0.4268	1.7359	0.5260	-0.0908	Unreliable
SARIMA(7,0,7)(0,1,1)53_CSS	-29.3094	808.2122	517.7657	-0.1571	1.7985	0.5432	-0.0465	Min(ME, MPE)
SARIMA(0,0,4)(0,1,1)53_CSS	159.6247	861.4001	563.5225	0.5084	1.9498	0.5912	-0.0191	
SARIMA(1,0,4)(0,1,1)53_CSS	147.3068	860.1077	560.2097	0.4641	1.9381	0.5877	-0.0263	
SARIMA(0,0,5)(0,1,1)53_CSS	151.7008	859.8969	564.7062	0.4815	1.9543	0.5924	-0.0323	
SARIMA(1,0,5)(0,1,1)53_CSS	148.9339	859.5103	561.1488	0.4701	1.9412	0.5887	-0.0309	
SARIMA(0,0,7)(0,1,1)53_CSS	154.9233	859.7620	563.8502	0.4927	1.9511	0.5915	-0.0348	
SARIMA(4,0,4)(0,1,1)53_CSS	151.2835	859.7354	559.3019	0.4718	1.9341	0.5868	-0.0366	
SARIMA(7,0,4)(0,1,1)53_CSS	-21.4886	793.2256	524.0707	-0.1237	1.8260	0.5498	-0.0537	Min(ACF1)
SARIMA(8,0,8)(0,1,1)53_CSSML	118.5706	655.5543	440.5241	0.3720	1.5232	0.4621	-0.0288	Unreliable
SARIMA(7,0,7)(0,1,1)53_CSSML	67.0606	656.0991	432.1266	0.1926	1.4964	0.4533	0.0052	
SARIMA(0,0,4)(0,1,1)53_CSSML	137.2562	689.6776	457.4647	0.4349	1.5822	0.4799	-0.0112	
SARIMA(1,0,4)(0,1,1)53_CSSML	125.3414	687.4188	458.6170	0.3942	1.5864	0.4811	-0.0264	
SARIMA(0,0,5)(0,1,1)53_CSSML	129.4646	687.5947	459.0999	0.4085	1.5879	0.4816	-0.0270	
SARIMA(1,0,5)(0,1,1)53_CSSML	75.6108	698.4575	466.0015	0.2194	1.6134	0.4889	0.0096	Unreliable
SARIMA(0,0,7)(0,1,1)53_CSSML	127.7971	687.4746	458.3763	0.4028	1.5855	0.4809	-0.0273	
SARIMA(4,0,4)(0,1,1)53_CSSML	118.0432	685.2040	456.8205	0.3687	1.5800	0.4792	-0.0239	
SARIMA(7,0,4)(0,1,1)53_CSSML	69.1422	689.2300	464.1822	0.1983	1.6066	0.4870	-0.0039	

Figure 11: Error measures of all models with comments on reliability, model with minimum errors and BIC/AIC score

to its estimate's unreliability, the next best model should be chosen as the optimal models in terms of error scores. This model was SARIMA(7,0,7)$(0,1,1)_{53}$ estimated using ML which has the lowest RMSE, MAE, MAPE, and MASE. This is expected as models with more parameter tends to produce lower error scores at the risk of overfitting. Note that the same model, computed using CSS, yielded the lowest ME and MPE. However, ML is generally considered more efficient and consistent than CSS in parameter estimates. Additionally, the ML-estimated model allows for BIC and AIC scores, which lends it further validity. Note that although the ML-estimated SARIMA(7,0,7)$(0,1,1)_{53}$ ranked in the middle in terms of AIC or BIC scores, the models above it were all some variants of SARIMA(0,0,4)$(0,1,1)_{53}$ with one or another over-parametrised parameter. Naturally, AIC and BIC prefer the more parsimonious models over the higher-order models. Thus, this report will proceed with the ML-estimated SARIMA(7,0,7)$(0,1,1)_{53}$ in further analysis and comparison against other models. The ML-estimated SARIMA(0,0,4)$(0,1,1)_{53}$ resulted in only around 6% difference in RMSE, MAE, MAPE, and MASE, to SARIMA(7,0,7)$(0,1,1)_{53}$.

Moreover, it is preferred by BIC and AIC scores. Hence, it is also selected for further considerations. Another model is also selected along with SARIMA(7,0,7)$(0,1,1)_{53}$ (ML-estimated) and SARIMA(0,0,4)$(0,1,1)_{53}$ (ML-estimated). This model had the lowest residual lag 1 autocorrelation (ACF1) and is SARIMA(7,0,4)$(0,1,1)_{53}$ (CSS), which coincidentally, also had the greatest number of significant coefficients.

3.3. Residual Analysis

5 diagnostic tools were used to evaluate the performance of the chosen models and to check whether important assumptions are met. The residuals are checked for randomness, normality (including qq-plot) as well as existing correlation using the ACF plot and the Ljung-Box test.

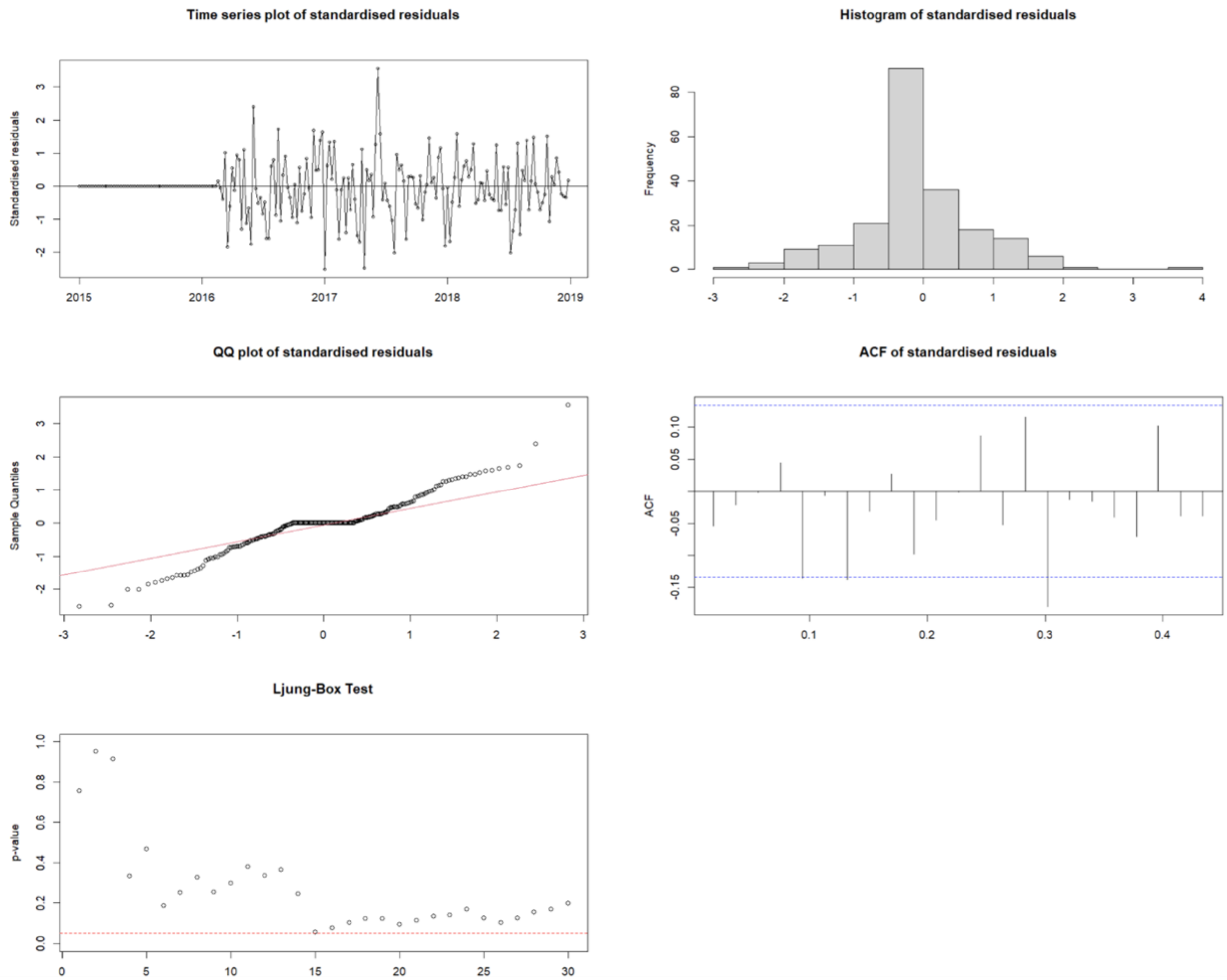


Figure 12: Diagnostic plots of SARIMA(7,0,4)(0,1,1)53 (CSS)

Figure 12 illustrates the diagnostics plots of the model SARIMA(7,0,4)(0,1,1)$_{53}$ estimated using the CSS method. The standardised residual plot appears to be random and does not exhibit any discernible pattern. However, there is an observed outlier in the residuals which is in mid 2017. This outlier may affect the normality of the plot. Despite this, the histogram of the standardised residuals appears to resemble standard normality but with more data distributed around the mean than is usually seen in a normally distributed histogram. Similarly, although many residuals lie along the reference line of the QQ-plot, the tails of the plot seem to deviate considerably. Consequently, the Shapito-Wilk test statistic for the residual of this model was 0.943, p-value < 0.05. There is statistically significant evidence to suggest that the residual of the model is not normally distributed. Additionally, there other signs that that SARIMA(7,0,4)(0,1,1)$_{53}$

(CSS) was not the optimal model for the current objective. The ACF plot shows near significant autocorrelation at a few lag and one clearly significant autocorrelation. The Ljung box test also reveals a lag with significant autocorrelation which violates the model's assumption. SARIMA(7,0,4)$(0,1,1)_{53}$ (CSS) will not be selected for the forecasting task of the analysis.

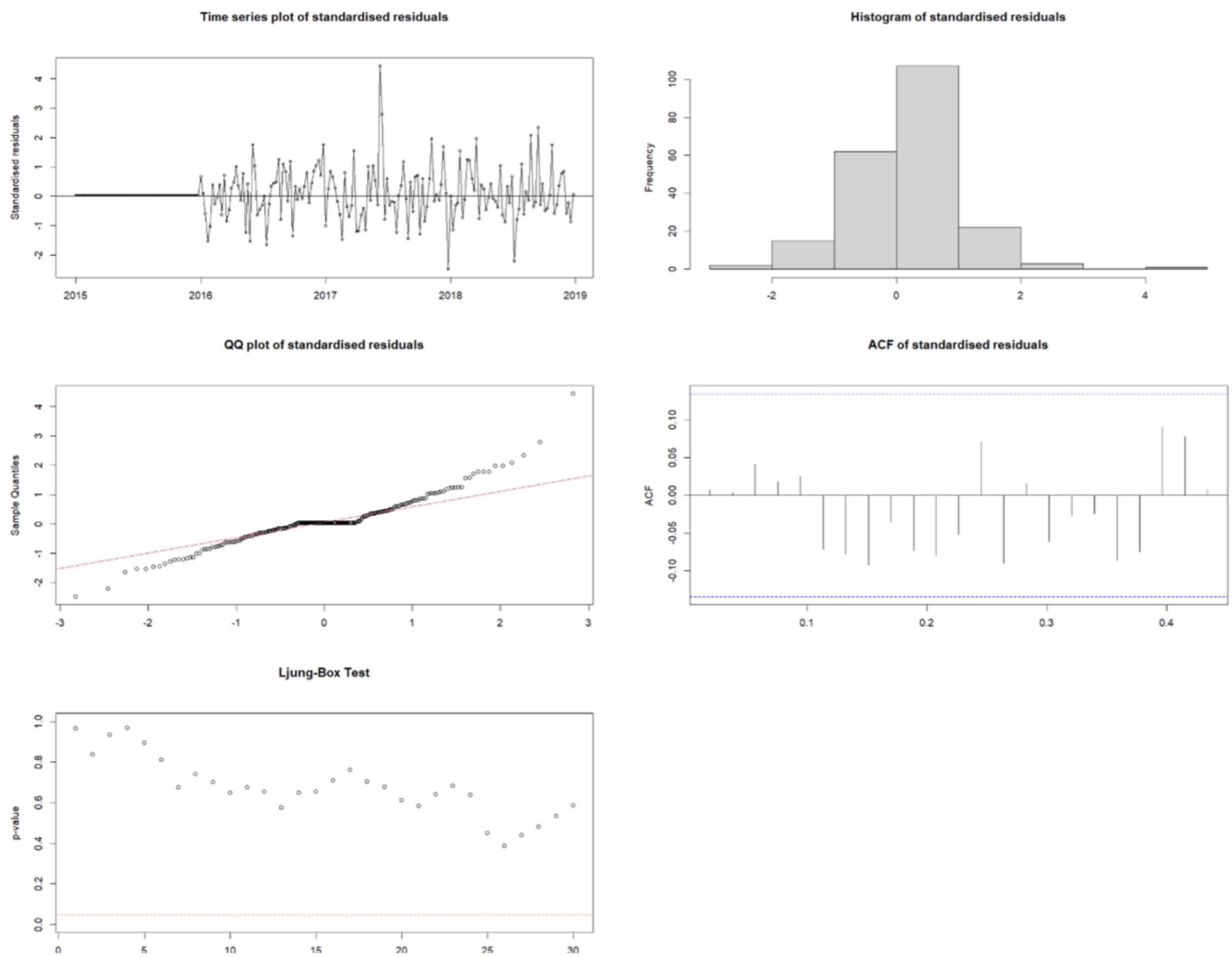


Figure 13:Diagnostic plots of SARIMA(7,0,7)(0,1,1)53 (ML)

The figure above illustrates the diagnostics of the residuals of the ML-estimated model SARIMA(7,0,7)$(0,1,1)_{53}$. Although the one outlier still exists, there is a significant improvement on the standardised residual plot. Specifically, the residuals appear to have a more constant variance with no apparent trend, resembling a white noise series with no clear patterns or behaviour. The histogram also shwos that the residuals appear normally distributed. Corroborating with this is the fact that the QQ-plot now shows less points deviating from the reference line, a promising sign of normality. The test for normality yielded a test statistic of 0.9287, p-value < 0.05. Although the test rejects the hypothesis that the residual is normally distributed, it may be largely due to the outlier. Moreover, there are no residual autocorrelation left as seen in both the ACF plot and the Ljung Box test. Therefore, this model may be safely used to forecast future energy consumption load.

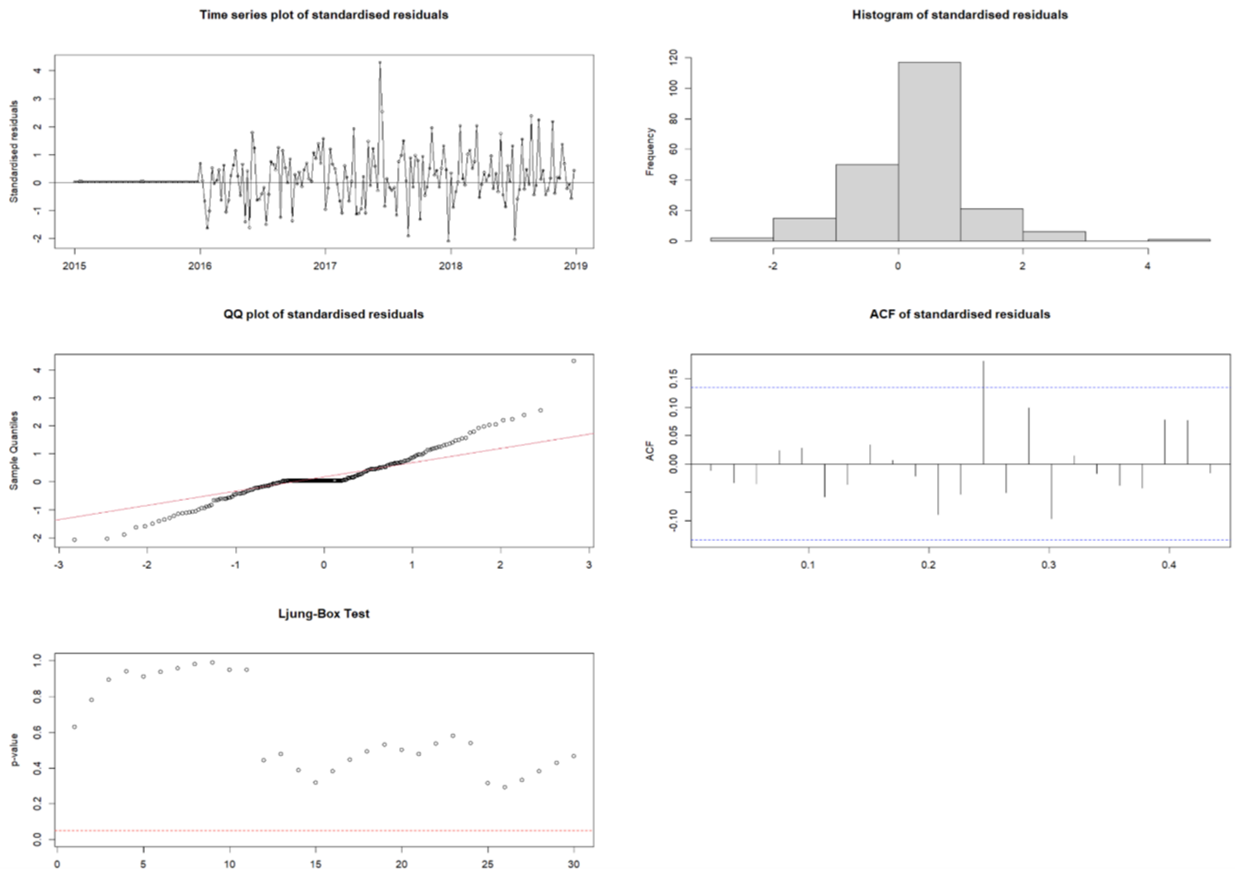


Figure 14:Diagnostic plots of SARIMA(7,0,4)(0,1,1)53 (CSS)

The diagnostics of the ML-estimated model SARIMA(0,0,4)(0,1,1)$_{53}$ are similar to that of the high-order model, SARIMA(7,0,7)(0,1,1)$_{53}$. The standardised residuals appear random with the exceptions of the same outlier and a few high values. The distribution of the residuals, according to both the histogram and the QQ-plot, aligns well with normality. The Shapiro-wilk test statistic was 0.93305, p-value < 0.05. Again, the statistical test rejects the hypothesis that the distribution is normal, but the outlier may still be a significant factor. On the other hand, the residual ACF plot of this model shows 1 high autocorrelation. However, the Ljung Box test did not flag any autocorrelation as high. Therefore, this model is still safe for the task. In fact, the ML-estimated SARIMA(0,0,4)(0,1,1)$_{53}$ may be preferred in practice as a more parsimonious model.

3.4. Forecasting

SAIRMA(0,0,4)(0,1,1)53					
Period	Point Forecast	Low 80	High 80	Low 95	High 95
2019-Week 1	27113.29	25948.16	28278.42	25331.37	28895.2
2019-Week 2	29015.11	27785.77	30244.46	27135	30895.23
2019-Week 3	31615.05	30385.71	32844.38	29734.94	33495.15
2019-Week 4	31239.72	30007.82	32471.61	29355.69	33123.74
2019-Week 5	29521.29	28251.37	30791.2	27579.12	31463.45
2019-Week 6	27959.01	26689.1	29228.93	26016.85	29901.18
2019-Week 7	30464.37	29194.46	31734.29	28522.21	32406.54
2019-Week 8	30194.65	28924.73	31464.56	28252.48	32136.81
2019-Week 9	29538.87	28268.95	30808.78	27596.7	31481.03
2019-Week 10	28310.72	27040.8	29580.63	26368.55	30252.88

SAIRMA(7,0,7)(0,1,1)53					
Period	Point Forecast	Low 80	High 80	Low 95	High 95
2019-Week 1	27194.71	26056.6	28332.82	25454.12	28935.29
2019-Week 2	29488.49	28273.33	30703.66	27630.07	31346.92
2019-Week 3	31903.89	30687.18	33120.6	30043.09	33764.69
2019-Week 4	32040.1	30824.16	33256.03	30180.49	33899.71
2019-Week 5	29881.48	28633.47	31129.5	27972.81	31790.16
2019-Week 6	28407.48	27157.95	29657.01	26496.49	30318.46
2019-Week 7	30541.3	29291.75	31790.84	28630.28	32452.32
2019-Week 8	30655.86	29407.02	31904.71	28745.92	32565.81
2019-Week 9	30137.96	28874.46	31401.45	28205.61	32070.3
2019-Week 10	28792.97	27529.06	30056.88	26859.98	30725.96

Figure 15: Point estimates of chosen models with 80% and 95% confidence interval

Figure 15 shows the point estimates with 80% and 95% confidence intervals, of the ML-estimated SARIMA(0,0,4)(0,1,1)$_{53}$ and SARIMA(7,0,7)(0,1,1)$_{53}$ models. SARIMA(7,0,7)(0,1,1)$_{53}$ models. The predictions of both models were similar for the first prediction week. Afterwards, the predictions were different, in general, around 300 to 500 megawatts. On the other hand, the predicted confidence intervals are more consistent between the models, especially at the 95% intervals. This likely indicates that the two models are effectively comparable in their predictive power.

Evidently, figure 16 (main time series plots with predicted values) shows that the two models predicted values that are closely aligned with the patterns of the original time series. Particularly, the estimates exhibit the same recovery pattern seen at the start of the year. Given the current data, the total load in first week of each year is always observed to be low, while the load greatly increases in the second week. This suggests that the model is effective in capturing the seasonal as well as the moving average behaviours of the time series.

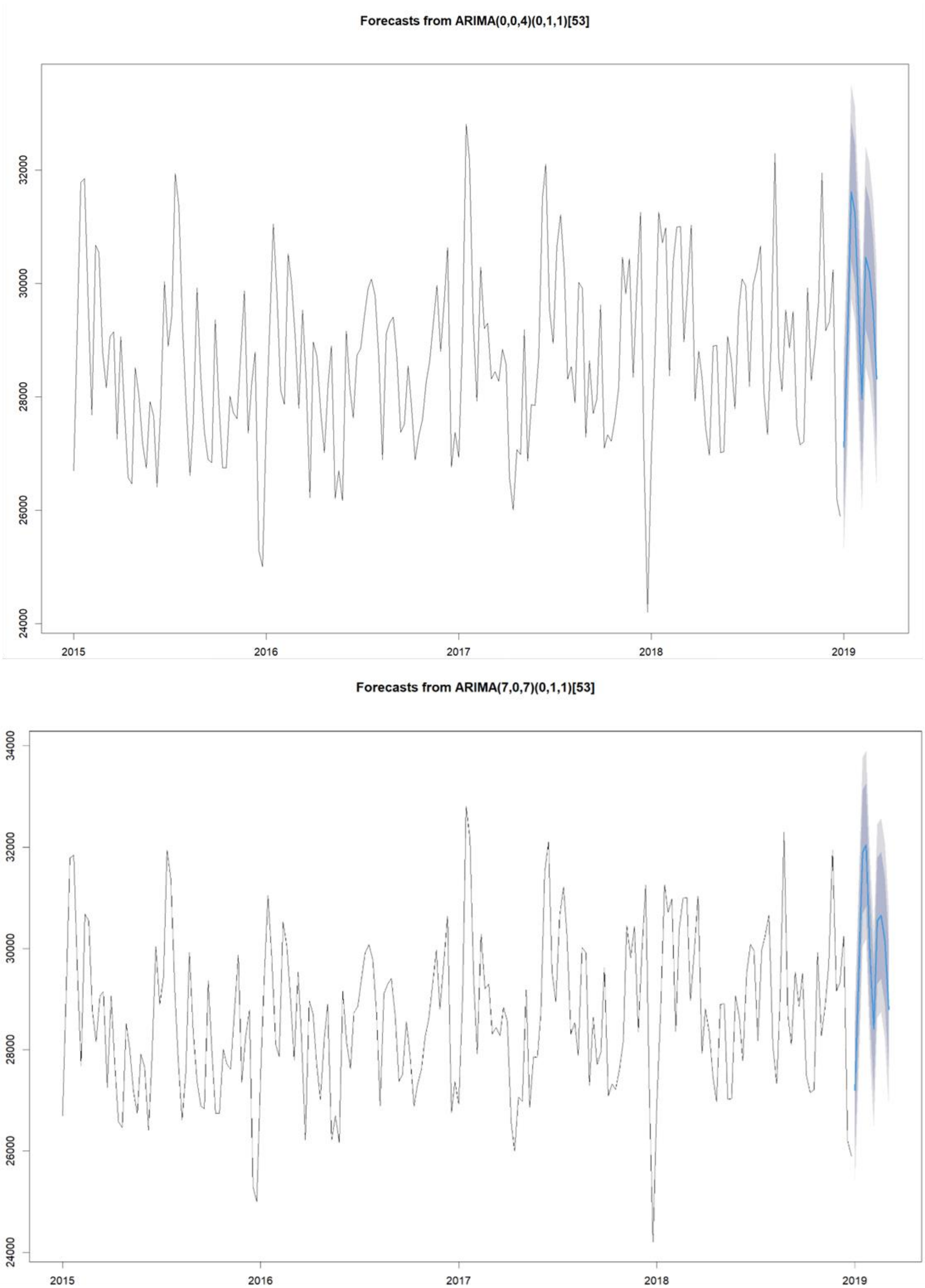


Figure 16: Time series of total consumption load with forecasted values for 2019

4. Conclusion

This paper aims to use the seasonal auto-regressive moving average model (SARIMA) to analyse and predict weekly aggregated data for the total energy load using a dataset of daily total energy load of the four largest cities in Spain. The original time series exhibits stationarity, seasonality and potential moving average and autoregressive behaviour. The time series also lacks clear changing variance or change point. The initial ACF and PACF plots show that there was seasonal trend in the model. Therefore, a one-order of seasonal differencing was applied to the series. After removing the seasonal trend, the series appear to have one significant order of moving average component but no significant autoregressive components.

Using a combination of ACF, PACF, EACF, and BIC table, 9 models were selected and fitted to the data. Based on the BIC and AIC scores computed for each model, SARIMA$(0,0,4)(0,1,1)_{53}$ was preferred. The error measures, however, were lowest for SARIMA$(7,0,7)(0,1,1)_{53}$ the highest-order model of the reliable models. In keeping with parsimony, SARIMA$(0,0,4)(0,1,1)_{53}$ is preferred as it is effective with less parameters. Standard procedure also calls for the selected models to be overparametrised to thoroughly explore the potential model space. However, the overparametrised models of SARIMA$(0,0,4)(0,1,1)_{53}$ was also selected during the model fitting and was expected to perform worse than the current best model.

Overall, the two models had similar prediction intervals for the total energy load in the next 10 weeks. Both models predicted that within the next 10 weeks, the total energy consumed is between 2500 to 3000 MW.

Several caveats are to be considered although the forecasts appear to simulate the pattern of the time series well. First, the paper only considers possible models using SARIMA. Statistical tests seem to suggest the model is trend stationary, although not difference stationary. This means the analysis may benefit from detrending before implementing SARIMA. Additionally, this paper did not consider other possible seasonal frequency beyond the annual cycle. Further studies should consider and incoporate these findings as well as corroborate the forecasted values with the actual values at later dates.

Appendix A: Coefficient Tests of SARIMA Models

```
--- SARIMA(8,0,8)(0,1,1)53 (CSS) ---

z test of coefficients:

       Estimate Std. Error  z value  Pr(>|z|)    
ar1  -0.0461856  0.0071485  -6.4608 1.041e-10 ***
ar2  -0.0376321  0.0037118 -10.1384 < 2.2e-16 ***
ar3  -0.6014631  0.0074187 -81.0734 < 2.2e-16 ***
ar4  -0.6677240  0.0089896 -74.2774 < 2.2e-16 ***
ar5   0.1605138  0.0101277  15.8490 < 2.2e-16 ***
ar6  -0.0833396  0.0083032 -10.0370 < 2.2e-16 ***
ar7  -0.1134792  0.0145920  -7.7768 7.438e-15 ***
ar8   0.0801609  0.0177437   4.5177 6.251e-06 ***
ma1   0.3905203  0.0217082  17.9895 < 2.2e-16 ***
ma2   0.1303496        NaN      NaN       NaN    
ma3   0.8935986        NaN      NaN       NaN    
ma4   1.4424554  0.0350685  41.1325 < 2.2e-16 ***
ma5   0.3129386  0.0624982   5.0072 5.524e-07 ***
ma6   0.2116294        NaN      NaN       NaN    
ma7   0.5131512  0.0177106  28.9743 < 2.2e-16 ***
ma8   0.3576143        NaN      NaN       NaN    
sma1 -0.6723666  0.0879068  -7.6486 2.031e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
--- SARIMA(8,0,8)(0,1,1)53 (ML) ---

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)    
ar1  -0.512897        NaN     NaN      NaN    
ar2  -0.255648   0.171790 -1.4881  0.13671    
ar3   0.138303   0.212720  0.6502  0.51559    
ar4   1.029386   0.124526  8.2664  < 2e-16 ***
ar5   0.755952        NaN     NaN      NaN    
ar6   0.124107   0.082806  1.4988  0.13393    
ar7   0.068490   0.142570  0.4804  0.63095    
ar8  -0.350575   0.179989 -1.9478  0.05144 .  
ma1   0.911969        NaN     NaN      NaN    
ma2   0.471041   0.266838  1.7653  0.07752 .  
ma3   0.024806   0.207014  0.1198  0.90462    
ma4  -0.847170   0.094010 -9.0115  < 2e-16 ***
ma5  -1.055708        NaN     NaN      NaN    
ma6  -0.254348        NaN     NaN      NaN    
ma7  -0.123495        NaN     NaN      NaN    
ma8   0.155673   0.229656  0.6779  0.49787    
sma1 -0.950915        NaN     NaN      NaN    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
--- SARIMA(8,0,8)(0,1,1)53 (CSS-ML) ---

z test of coefficients:

      Estimate Std. Error z value  Pr(>|z|)    
ar1  -0.326162   0.285492 -1.1425 0.2532653    
ar2  -0.011901   0.227356 -0.0523 0.9582554    
ar3  -0.713274   0.251853 -2.8321 0.0046242 ** 
ar4  -0.705821   0.259658 -2.7183 0.0065625 ** 
ar5   0.189214   0.228002  0.8299 0.4066079    
ar6  -0.150051   0.219311 -0.6842 0.4938536    
ar7   0.018152   0.208217  0.0872 0.9305315    
ar8   0.090524   0.224945  0.4024 0.6873697    
ma1   0.696325   0.279898  2.4878 0.0128543 *  
ma2   0.141794   0.248362  0.5709 0.5680553    
ma3   0.845992   0.281448  3.0059 0.0026483 ** 
ma4   1.405323   0.313806  4.4783 7.523e-06 ***
ma5   0.264912   0.304049  0.8713 0.3836013    
ma6   0.163922   0.266518  0.6150 0.5385224    
ma7   0.335440   0.249753  1.3431 0.1792438    
ma8   0.297030   0.215228  1.3801 0.1675638    
sma1 -0.996771   0.291824 -3.4157 0.0006363 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
--- SARIMA(7,0,7)(0,1,1)53 (CSS) ---

z test of coefficients:

      Estimate Std. Error z value  Pr(>|z|)    
ar1  -0.103399   0.135793 -0.7614  0.446394    
ar2   0.767637   0.123239  6.2288 4.699e-10 ***
ar3   0.596479   0.095167  6.2677 3.664e-10 ***
ar4   0.009039   0.147251  0.0614  0.951053    
ar5  -0.229173   0.137019 -1.6726  0.094413 .  
ar6  -0.152131   0.139538 -1.0903  0.275603    
ar7   0.105019   0.116315  0.9029  0.366586    
ma1   0.464067   0.146460  3.1686  0.001532 ** 
ma2  -0.788527   0.122574 -6.4331 1.250e-10 ***
ma3  -0.949991   0.182808 -5.1967 2.029e-07 ***
ma4  -0.049344   0.210243 -0.2347  0.814443    
ma5   0.243185   0.152336  1.5964  0.110406    
ma6   0.113885   0.104391  1.0909  0.275297    
ma7  -0.150602   0.113925 -1.3219  0.186186    
sma1 -0.737655   0.092743 -7.9538 1.809e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
--- SARIMA(7,0,7)(0,1,1)53 (ML) ---

z test of coefficients:

      Estimate Std. Error z value  Pr(>|z|)    
ar1  -0.696885   0.408457 -1.7061 0.0879819 .  
ar2  -0.039401   0.359031 -0.1097 0.9126128    
ar3  -0.179753   0.209530 -0.8579 0.3909535    
ar4   0.191680   0.124840  1.5354 0.1246850    
ar5   0.763662   0.117695  6.4885 8.669e-11 ***
ar6   0.579478   0.428884  1.3511 0.1766539    
ar7   0.377311   0.233657  1.6148 0.1063523    
ma1   1.077238   0.388579  2.7723 0.0055670 ** 
ma2   0.262428   0.469707  0.5587 0.5763622    
ma3   0.192825   0.190005  1.0148 0.3101798    
ma4   0.154534   0.132967  1.1622 0.2451574    
ma5  -0.732915   0.199473 -3.6743 0.0002385 ***
ma6  -0.918845   0.404073 -2.2740 0.0229685 *  
ma7  -0.536200   0.384968 -1.3928 0.1636675    
sma1 -0.986888        NaN     NaN       NaN    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
--- SARIMA(7,0,7)(0,1,1)53 (CSS-ML) ---

z test of coefficients:

      Estimate Std. Error z value  Pr(>|z|)    
ar1  -0.957298   0.521734 -1.8348 0.0665295 .  
ar2   0.468223   0.432778  1.0819 0.2792969    
ar3   0.329512   0.331534  0.9939 0.3202718    
ar4   0.294936   0.325377  0.9064 0.3647002    
ar5   0.896393   0.265531  3.3758 0.0007359 ***
ar6   0.194463   0.557588  0.3488 0.7272716    
ar7  -0.226229   0.239657 -0.9440 0.3451844    
ma1   1.335301   0.472746  2.8246 0.0047345 ** 
ma2  -0.186539   0.561333 -0.3323 0.7396519    
ma3  -0.541716   0.347912 -1.5570 0.1194592    
ma4  -0.115398   0.363378 -0.3176 0.7508107    
ma5  -0.777654   0.399541 -1.9464 0.0516104 .  
ma6  -0.660631   0.582645 -1.1338 0.2568576    
ma7  -0.023237   0.445495 -0.0522 0.9584014    
sma1 -0.959829   0.134989 -7.1104 1.157e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
--- SARIMA(0,0,4)(0,1,1)53 (CSS) ---

z test of coefficients:

      Estimate Std. Error z value  Pr(>|z|)    
ma1   0.386131   0.079863  4.8349 1.332e-06 ***
ma2   0.034097   0.086660  0.3935 0.6939850    
ma3   0.096116   0.085010  1.1306 0.2582074    
ma4   0.267216   0.069529  3.8433 0.0001214 ***
sma1 -0.699992   0.088267 -7.9304 2.184e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
--- SARIMA(0,0,4)(0,1,1)53 (ML) ---

z test of coefficients:

       Estimate Std. Error z value  Pr(>|z|)    
ma1   0.3458220  0.0773253  4.4723 7.738e-06 ***
ma2  -0.0049762  0.0821617 -0.0606  0.951705    
ma3   0.0738270  0.0793297  0.9306  0.352042    
ma4   0.2731633  0.0688422  3.9680 7.249e-05 ***
sma1 -0.9999543  0.3407952 -2.9342  0.003344 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
--- SARIMA(0,0,4)(0,1,1)53 (CSS-ML) ---

z test of coefficients:

       Estimate Std. Error z value  Pr(>|z|)    
ma1   0.3457574  0.0773308  4.4711 7.780e-06 ***
ma2  -0.0049408  0.0821549 -0.0601  0.952044    
ma3   0.0738290  0.0793297  0.9307  0.352029    
ma4   0.2731114  0.0688447  3.9671 7.276e-05 ***
sma1 -0.9999809  0.3408158 -2.9341  0.003345 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
--- SARIMA(1,0,4)(0,1,1)53 (CSS) ---

z test of coefficients:

      Estimate Std. Error z value  Pr(>|z|)    
ar1   0.155768   0.266565  0.5844    0.5590    
ma1   0.240701   0.259287  0.9283    0.3532    
ma2  -0.026922   0.131872 -0.2041    0.8382    
ma3   0.074093   0.083691  0.8853    0.3760    
ma4   0.272928   0.068661  3.9750 7.037e-05 ***
sma1 -0.688043   0.089435 -7.6933 1.434e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
--- SARIMA(1,0,4)(0,1,1)53 (ML) ---

z test of coefficients:

      Estimate Std. Error z value  Pr(>|z|)    
ar1   0.310495   0.288056  1.0779  0.281079    
ma1   0.054840   0.279065  0.1965  0.844208    
ma2  -0.110822   0.127477 -0.8694  0.384656    
ma3   0.056554   0.076229  0.7419  0.458152    
ma4   0.280690   0.070281  3.9938 6.502e-05 ***
sma1 -0.998091   0.345571 -2.8882  0.003874 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
--- SARIMA(1,0,4)(0,1,1)53 (CSS-ML) ---

z test of coefficients:

      Estimate Std. Error z value  Pr(>|z|)    
ar1   0.308520   0.288789  1.0683  0.285376    
ma1   0.056790   0.279784  0.2030  0.839153    
ma2  -0.110105   0.127750 -0.8619  0.388752    
ma3   0.056666   0.076247  0.7432  0.457364    
ma4   0.280670   0.070231  3.9964 6.432e-05 ***
sma1 -0.999883   0.346432 -2.8862  0.003899 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
--- SARIMA(0,0,5)(0,1,1)53 (CSS) ---

z test of coefficients:

      Estimate Std. Error z value  Pr(>|z|)    
ma1   0.401418   0.081968  4.8972 9.719e-07 ***
ma2   0.041222   0.088703  0.4647 0.6421304    
ma3   0.080924   0.087499  0.9249 0.3550392    
ma4   0.285385   0.073726  3.8709 0.0001084 ***
ma5   0.060179   0.080417  0.7483 0.4542533    
sma1 -0.693674   0.088619 -7.8276 4.972e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
--- SARIMA(0,0,5)(0,1,1)53 (ML) ---

z test of coefficients:

        Estimate   Std. Error z value  Pr(>|z|)    
ma1   0.36425372  0.07888522  4.6175 3.884e-06 ***
ma2   0.00020463  0.08329462  0.0025  0.998040    
ma3   0.05486963  0.08091961  0.6781  0.497724    
ma4   0.29579154  0.07289621  4.0577 4.956e-05 ***
ma5   0.07868632  0.08057612  0.9765  0.328794    
sma1 -0.99998753  0.35122938 -2.8471  0.004412 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
--- SARIMA(0,0,5)(0,1,1)53 (CSS-ML) ---

z test of coefficients:

        Estimate   Std. Error z value  Pr(>|z|)    
ma1   0.36426906  0.07888519  4.6177 3.880e-06 ***
ma2   0.00021169  0.08329392  0.0025  0.997972    
ma3   0.05486003  0.08092047  0.6779  0.497803    
ma4   0.29578000  0.07289825  4.0574 4.961e-05 ***
ma5   0.07871091  0.08057531  0.9769  0.328638    
sma1 -0.99989787  0.35118583 -2.8472  0.004411 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
--- SARIMA(1,0,5)(0,1,1)53 (CSS) ---

z test of coefficients:

      Estimate Std. Error z value  Pr(>|z|)    
ar1  -0.179303   0.674417 -0.2659  0.790344    
ma1   0.578414   0.673873  0.8583  0.390703    
ma2   0.106926   0.286929  0.3727  0.709405    
ma3   0.080257   0.095558  0.8399  0.400976    
ma4   0.305709   0.096687  3.1618  0.001568 ** 
ma5   0.107781   0.203023  0.5309  0.595500    
sma1 -0.688702   0.089249 -7.7166 1.195e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
--- SARIMA(1,0,5)(0,1,1)53 (ML) ---

z test of coefficients:

      Estimate Std. Error z value  Pr(>|z|)    
ar1   0.999552        NaN     NaN       NaN    
ma1  -0.665480   0.072770 -9.1450 < 2.2e-16 ***
ma2  -0.363864   0.093603 -3.8873 0.0001014 ***
ma3   0.076047   0.102504  0.7419 0.4581536    
ma4   0.221211   0.083461  2.6505 0.0080377 ** 
ma5  -0.227837   0.073515 -3.0992 0.0019405 ** 
sma1 -0.903556   0.150505 -6.0035 1.931e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


--- SARIMA(1,0,5)(0,1,1)53 (CSS-ML) ---

z test of coefficients:

      Estimate Std. Error z value  Pr(>|z|)    
ar1   0.999504        NaN     NaN       NaN    
ma1  -0.673132   0.076463 -8.8034 < 2.2e-16 ***
ma2  -0.347832   0.095017 -3.6607 0.0002515 ***
ma3   0.074247   0.102314  0.7257 0.4680369    
ma4   0.219267   0.084763  2.5868 0.0096861 ** 
ma5  -0.228045   0.073570 -3.0997 0.0019371 ** 
sma1 -0.944510   0.119266 -7.9193 2.388e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


--- SARIMA(0,0,7)(0,1,1)53 (CSS) ---

z test of coefficients:

      Estimate Std. Error z value  Pr(>|z|)    
ma1   0.404022   0.084177  4.7997 1.589e-06 ***
ma2   0.041311   0.089131  0.4635  0.643018    
ma3   0.079020   0.087718  0.9008  0.367673    
ma4   0.282893   0.074556  3.7944  0.000148 ***
ma5   0.051904   0.090703  0.5722  0.567158    
ma6  -0.013942   0.099267 -0.1405  0.888302    
ma7  -0.015542   0.087478 -0.1777  0.858987    
sma1 -0.694193   0.088794 -7.8181 5.364e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
--- SARIMA(0,0,7)(0,1,1)53 (ML) ---

z test of coefficients:

       Estimate Std. Error z value  Pr(>|z|)    
ma1   0.3657077  0.0803498  4.5514 5.328e-06 ***
ma2   0.0010357  0.0834144  0.0124  0.990094    
ma3   0.0546288  0.0838079  0.6518  0.514508    
ma4   0.2957232  0.0744256  3.9734 7.085e-05 ***
ma5   0.0857059  0.0892564  0.9602  0.336944    
ma6   0.0186022  0.0939347  0.1980  0.843019    
ma7  -0.0025481  0.0829223 -0.0307  0.975486    
sma1 -0.9998057  0.3485577 -2.8684  0.004125 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


--- SARIMA(0,0,7)(0,1,1)53 (CSS-ML) ---

z test of coefficients:

       Estimate Std. Error z value  Pr(>|z|)    
ma1   0.3657115  0.0803543  4.5512 5.333e-06 ***
ma2   0.0010848  0.0834146  0.0130  0.989624    
ma3   0.0546115  0.0838034  0.6517  0.514619    
ma4   0.2957041  0.0744238  3.9732 7.090e-05 ***
ma5   0.0856548  0.0892540  0.9597  0.337219    
ma6   0.0185428  0.0939334  0.1974  0.843512    
ma7  -0.0025937  0.0829206 -0.0313  0.975047    
sma1 -0.9999754  0.3486369 -2.8682  0.004128 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


--- SARIMA(4,0,4)(0,1,1)53 (CSS) ---

z test of coefficients:

      Estimate Std. Error z value  Pr(>|z|)    
ar1   0.360537   0.317096  1.1370    0.2555    
ar2  -0.198073   0.263574 -0.7515    0.4524    
ar3   0.251557   0.298369  0.8431    0.3992    
ar4   0.027916   0.276261  0.1011    0.9195    
ma1   0.032773   0.315807  0.1038    0.9173    
ma2   0.069323   0.226064  0.3067    0.7591    
ma3  -0.114863   0.266184 -0.4315    0.6661    
ma4   0.158995   0.211072  0.7533    0.4513    
sma1 -0.664804   0.092160 -7.2136 5.451e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
--- SARIMA(4,0,4)(0,1,1)53 (ML) ---

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1   0.047378   0.279287  0.1696 0.865294
ar2   0.016564   0.292253  0.0567 0.954804
ar3  -0.021596   0.245468 -0.0880 0.929894
ar4   0.209727   0.174635  1.2009 0.229772
ma1   0.316690   0.282278  1.1219 0.261903
ma2  -0.030255   0.309192 -0.0979 0.922051
ma3   0.079635   0.291864  0.2729 0.784968
ma4   0.120928   0.170672  0.7085 0.478609
sma1 -0.999814   0.345488 -2.8939 0.003805 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

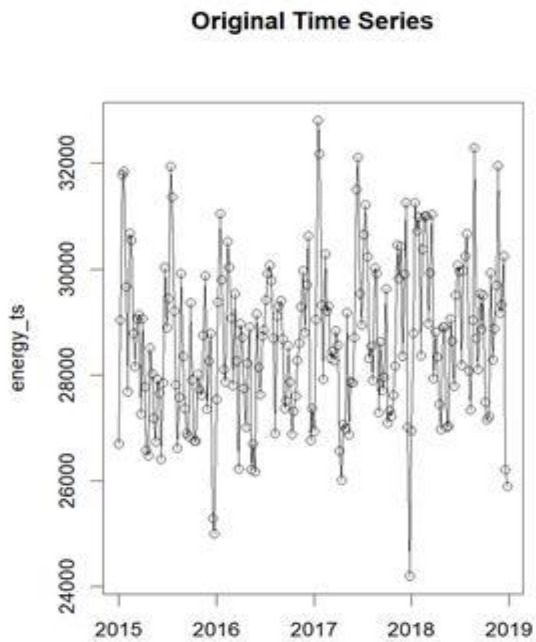
```
--- SARIMA(4,0,4)(0,1,1)53 (CSS-ML) ---

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1   0.046473   0.279580  0.1662 0.867981
ar2   0.013662   0.293022  0.0466 0.962812
ar3  -0.022047   0.246449 -0.0895 0.928716
ar4   0.209872   0.174359  1.2037 0.228713
ma1   0.317626   0.282662  1.1237 0.261143
ma2  -0.026946   0.309897 -0.0870 0.930709
ma3   0.081253   0.293225  0.2771 0.781703
ma4   0.120927   0.170258  0.7103 0.477546
sma1 -0.999881   0.345597 -2.8932 0.003813 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

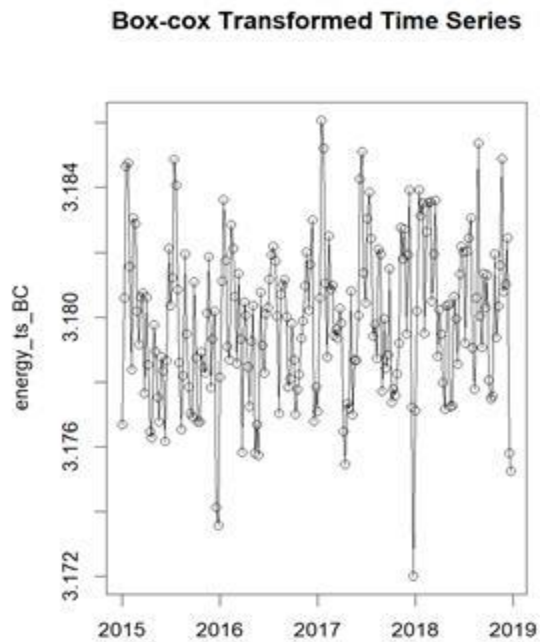
```
--- SARIMA(7,0,4)(0,1,1)53 (CSS) ---

z test of coefficients:

      Estimate Std. Error  z value  Pr(>|z|)
ar1   0.863363   0.027509  31.3853 < 2.2e-16 ***
ar2  -0.322095   0.027236 -11.8262 < 2.2e-16 ***
ar3   1.046877   0.023851  43.8926 < 2.2e-16 ***
ar4  -0.697505   0.070448  -9.9009 < 2.2e-16 ***
ar5   0.266578   0.036174   7.3693 1.716e-13 ***
ar6  -0.120779   0.036913  -3.2720  0.001068 **
ar7  -0.039861   0.072312  -0.5512  0.581475
ma1  -0.481613   0.042359 -11.3697 < 2.2e-16 ***
ma2  -0.037658   0.017581  -2.1420  0.032196 *
ma3  -1.067231   0.019112 -55.8421 < 2.2e-16 ***
ma4   0.526766   0.048601  10.8387 < 2.2e-16 ***
sma1 -0.712225   0.087229  -8.1650 3.214e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
--- SARIMA(7,0,4)(0,1,1)53 (ML) ---

z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)   
ar1  -0.474504   0.282713 -1.6784 0.093271 . 
ar2  -1.006452   0.360238 -2.7939 0.005208 **
ar3   0.078010   0.295026  0.2644 0.791457   
ar4  -0.246438   0.286812 -0.8592 0.390213   
ar5   0.426276   0.164127  2.5972 0.009398 **
ar6   0.147683   0.133745  1.1042 0.269502   
ar7   0.031732   0.132395  0.2397 0.810579   
ma1   0.849538   0.275641  3.0820 0.002056 **
ma2   1.226486   0.423890  2.8934 0.003811 **
ma3   0.382638   0.434304  0.8810 0.378297   
ma4   0.600776   0.327228  1.8360 0.066364 . 
sma1 -0.959636   0.351777 -2.7280 0.006373 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
--- SARIMA(7,0,4)(0,1,1)53 (CSS-ML) ---

z test of coefficients:

      Estimate Std. Error  z value  Pr(>|z|)    
ar1  -0.272289   0.380475  -0.7157   0.47420    
ar2   0.128781   0.104010   1.2382   0.21565    
ar3   0.955209   0.085498  11.1723 < 2.2e-16 ***
ar4   0.364433   0.339018   1.0750   0.28239    
ar5   0.031061   0.108654   0.2859   0.77498    
ar6  -0.016908   0.091273  -0.1852   0.85303    
ar7  -0.190406   0.081063  -2.3489   0.01883 *  
ma1   0.626510   0.386607   1.6205   0.10512    
ma2  -0.065198   0.098298  -0.6633   0.50716    
ma3  -1.046903   0.085055 -12.3086 < 2.2e-16 ***
ma4  -0.433618   0.372096  -1.1653   0.24388    
sma1 -0.925173   0.162134  -5.7062 1.155e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Appendix B: Original and Box-Cox-Transformed Time Series Plots for Weekly Load





Appendix C: R codes

```
rm(list=ls())
library(TSA)
library(tidyr)
library(tseries)
library(dplyr)
library(ggplot2)
library(ggrepel)
library(patchwork)
library(lmtest)
library(forecast)
library(fUnitRoots)
library(magrittr)
library(lubridate)
library(FitAR)

#============functions=============

#stationarity
stationarity <- function(ts) {
  ad <- adf.test(ts)
  pp <- pp.test(ts, alternative = c("stationary"))
  kpss <- kpss.test(ts)
  print(ad)
  print(pp)
  print(kpss)
}

#sort score function
sort.score <- function(x, score = c("bic", "aic")){
  if (score == "aic"){
    x[with(x, order(AIC)),]
  } else if (score == "bic") {
    x[with(x, order(BIC)),]
  } else {
    warning('score = "x" only accepts valid arguments ("aic","bic")')
  }
}

residual.analysis <- function(model, std = TRUE,start = 2, class =
c("ARIMA","GARCH","ARMA-GARCH", "fGARCH")[1]){
  library(TSA)
  library(FitAR)
  if (class == "ARIMA"){
    if (std == TRUE){
      res.model = rstandard(model)
    }else{
      res.model = residuals(model)
    }
  }else if (class == "GARCH"){
    res.model = model$residuals[start:model$n.used]
  }else if (class == "ARMA-GARCH"){
```

```
    res.model = model@fit$residuals
  }else if (class == "fGARCH"){
    res.model = model@residuals
  }else {
    stop("The argument 'class' must be either 'ARIMA' or 'GARCH' ")
  }
  par(mfrow=c(3,2))

  plot(res.model,type='o',ylab='Standardised residuals', main="Time
series plot of standardised residuals")
  abline(h=0)
  hist(res.model,main="Histogram of standardised residuals")
  qqnorm(res.model,main="QQ plot of standardised residuals")
  qqline(res.model, col = 2)
  seasonal_acf(res.model,main="ACF of standardised residuals")
  print(shapiro.test(res.model))
  k=0
  LBQPlot(res.model, lag.max = 30, StartLag = k + 1, k = 0, SquaredQ =
FALSE)
  par(mfrow=c(1,1))
}

#=========seasonal function========
##################################################
# These functions are developed by               #
# MATH1318 students                              #
# Le Van Tra Tran and Tin Trung Pham             #
# in 2024. WE thank them for their contribution! #
##################################################

# Helper function ---------------------------------------------------
-----------------

helper <- function(class = c("acf", "pacf"), ...) {

  # Capture additional arguments
  params <- match.call(expand.dots = TRUE)
  params <- as.list(params)[-1]

  # Calculate ACF/PACF values
  if (class == "acf") {
    acf_values <- do.call(acf, c(params, list(plot = FALSE)))
  } else if (class == "pacf") {
    acf_values <- do.call(pacf, c(params, list(plot = FALSE)))
  }

  # Extract values and lags
  acf_data <- data.frame(
    Lag = as.numeric(acf_values$lag),
    ACF = as.numeric(acf_values$acf)
  )

  # Identify seasonal lags to be highlighted
```

```
  seasonal_lags <- acf_data$Lag %% 1 == 0

  # Plot ACF/PACF values
  if (class == "acf") {
    do.call(acf, c(params, list(plot = TRUE)))
  } else if (class == "pacf") {
    do.call(pacf, c(params, list(plot = TRUE)))
  }

  # Add colored segments for seasonal lags
  for (i in which(seasonal_lags)) {
    segments(x0 = acf_data$Lag[i], y0 = 0, x1 = acf_data$Lag[i], y1 =
acf_data$ACF[i], col = "red")
  }
}


# seasonal_acf ---------------------------------------------------------
-----

seasonal_acf <- function(...) {
  helper(class = "acf", ...)
}


# seasonal_pacf --------------------------------------------------------
-----

seasonal_pacf <- function(...) {
  helper(class = "pacf", ...)
}


#----------data preprosessing----------------

setwd("D:/Study/RMIT/Year 2/Sem 1/Time series/Assignment
3/energy_dataset.csv")

energy <- read.csv("energy_dataset.csv")

energy <- energy %>% select(time, total.load.actual)

energy %<>% mutate(
  time = ymd_hms(energy$time, tz = "Europe/Paris")
)

energy %<>% mutate(
  day = day(energy$time),
  week = week(energy$time),
  month = month(energy$time),
  year = year(energy$time)
```

```
)

energy2 <- energy %>%
  group_by(week, year)  %>%
  summarise(mean_load = mean(total.load.actual, na.rm = TRUE), .groups
= "drop") %>%
  arrange(year, week)

energy_ts <- ts(energy2$mean_load, start = c(2015,1), frequency = 53)

end(energy_ts)
energy_ts


plot(energy_ts, type = "o",
     ylab = "Total Actual Load",
     main='Time series plot of Actual Energy Consumed')

shapiro.test(energy_ts)

#seasonal pacf and acf
par(mfrow=c(1,2))
seasonal_acf(energy_ts, lag.max = 215, main = "ACF")
seasonal_pacf(energy_ts, lag.max = 215, main = "PACF")
par(mfrow=c(1,1))

# Seasonality and existence of trend are obvious from the ACF and PACF
plots

adf.test(energy_ts)
pp.test(energy_ts)
kpss.test(energy_ts)



#boxcox
BC <- BoxCox.ar(energy_ts) #,lambda = seq(-1, 0.5, 0.01) If you get an
error.
BC$ci
lambda <- BC$lambda[which(max(BC$loglike) == BC$loglike)]
lambda

energy_ts_BC = ((energy_ts^lambda)-1)/lambda

par(mfrow=c(1,2))
plot(energy_ts, type = "o")
plot(energy_ts_BC, type = "o")
par(mfrow=c(1,1))

#=========================================modelling===================
==========
#======================model 1=====================
```

```r
m.energy_SARIMA000_010 = Arima(energy_ts,
                   order=c(0,0,0),
                   seasonal=list(order=c(0,1,0),
                                 period=53))
res.m.energy_SARIMA000_010 <- residuals(m.energy_SARIMA000_010)

par(mfrow=c(1,1))
plot(res.m.energy_SARIMA000_010, type = 'o',
     main = "Time series of Residuals After First Seasonal
Differencing",
     ylab = "residuals", xlab = "Period")



par(mfrow=c(1,2))
seasonal_acf(res.m.energy_SARIMA000_010, lag.max = 215, main = "First
seasonal differenced ACF")
seasonal_pacf(res.m.energy_SARIMA000_010, lag.max = 215, main = "First
seasonal differenced PACF")
par(mfrow=c(1,1))

adf.test(energy_ts)
pp.test(energy_ts)
kpss.test(energy_ts)

#=======================model 2 001=====================
m.energy_SARIMA000_011 = Arima(energy_ts,
                               order=c(0,0,0),
                               seasonal=list(order=c(0,1,1),
                                             period=53))
res.m.energy_SARIMA000_011 <- residuals(m.energy_SARIMA000_011)
par(mfrow=c(1,1))
plot(res.m.energy_SARIMA000_011, type = 'o',
     main = "Time series of residuals (SARIMA(0,0,0)(0,1,1)53)",
     ylab = 'residual',
     xlab = 'period')


#=======================model 2 101=====================
m.energy_SARIMA000_111 = Arima(energy_ts,
                               order=c(0,0,0),
                               seasonal=list(order=c(1,1,1),
                                             period=53))
res.m.energy_SARIMA000_111 <- residuals(m.energy_SARIMA000_111)
par(mfrow=c(1,1))
plot(res.m.energy_SARIMA000_111, type = 'o',
     main = "Time series of residuals (SARIMA(0,0,0)(1,1,1)53)",
     ylab = 'residual',
     xlab = 'period')



par(mfrow=c(1,2))
plot(res.m.energy_SARIMA000_011, type = 'o',
```

```r
     main = "Time series of residuals (SARIMA(0,0,0)(0,1,1)53)",
     ylab = 'residual',
     xlab = 'period')

plot(res.m.energy_SARIMA000_111, type = 'o',
     main = "Time series of residuals (SARIMA(0,0,0)(1,1,1)53)",
     ylab = 'residual',
     xlab = 'period')
par(mfrow=c(1,1))


m.energy_SARIMA010_011 = Arima(energy_ts,
                               order=c(0,1,0),
                               seasonal=list(order=c(1,1,1),
                                             period=53))
res.m.energy_SARIMA010_011 <- residuals(m.energy_SARIMA010_011)
plot(res.m.energy_SARIMA010_011, type = 'o',
     main = "Time series of residuals (SARIMA(0,1,0)(1,1,1)53)",
     ylab = 'residual',
     xlab = 'period')

stationarity(res.m.energy_SARIMA000_011)



#=======================model specification=====================
par(mfrow=c(2,2))
seasonal_acf(res.m.energy_SARIMA000_011, lag.max = 215,
             main = 'ACF of SARIMA(0,0,0)(0,1,1)53 residuals')
seasonal_pacf(res.m.energy_SARIMA000_011, lag.max = 215,
              main = 'PACF of SARIMA(0,0,0)(0,1,1)53 residuals')

seasonal_acf(res.m.energy_SARIMA000_111, lag.max = 215,
             main = 'ACF of SARIMA(0,0,0)(1,1,1)53 residuals')
seasonal_pacf(res.m.energy_SARIMA000_111, lag.max = 215,
              main= 'PACF of SARIMA(0,0,0)(1,1,1)53 residuals')
par(mfrow=c(1,1))

eacf(res.m.energy_SARIMA000_011)

resBIC = armasubsets(y=res.m.energy_SARIMA000_011,
                     nar=7,
                     nma=7,y.name='p',ar.method='ols')

plot(resBIC, main = "")
mtext("BIC table of residuals after seasonal components", side = 3,
line = 5.5, cex = 1.2)

#===================possible models==========
P <- c(8,7,0,1,0,1,0,4,7)
D <- rep(0,length(P))
Q <- c(8,7,4,4,5,5,7,4,4)
```

```r
seasonal_P <- rep(0,length(P))
seasonal_D <- rep(1,length(D))
seasonal_Q <- rep(1,length(Q))


# Create empty lists to store models
models_css <- list()
models_ml <- list()
models_CSSML <- list()

# Loop through all combinations of P, D, Q
for (i in seq_along(P)) {

  #model name
  model_name <- paste0("SARIMA(", P[i], ",", D[i], ",", Q[i],
                     ")(0,1,1)53")

  # Build and store models using CSS and ML methods
  models_css[[model_name]] <- Arima(energy_ts,
                                  order = c(P[i], D[i], Q[i]),
                                  seasonal=list(order=c(0,1,1),
                                                period=53),
                                  method = "CSS")
  models_ml[[model_name]] <- Arima(energy_ts,
                                 order = c(P[i], D[i], Q[i]),
                                 seasonal=list(order=c(0,1,1),
                                               period=53),
                                 method = "ML")
  models_CSSML[[model_name]] <- Arima(energy_ts,
                                    order = c(P[i], D[i], Q[i]),
                                    seasonal=list(order=c(0,1,1),
                                                  period=53),
                                    method = "CSS-ML")

}

#Save models
save(file='SARIMA01153CSS.Rdata', models_css)
save(file='SARIMA01153ML.Rdata', models_ml)
save(file='SARIMA01153CSSML.Rdata', models_CSSML)
load(file='SARIMA01153CSS.Rdata')
load(file='SARIMA01153ML.Rdata')
load(file='SARIMA01153CSSML.Rdata')

# Loop over all model names and extract coefficients
# for (name in names(models_css)) {
#   cat("\n---", name, "(CSS) ---\n")
#   print(coeftest(models_css[[name]]))
#
#   cat("\n---", name, "(ML) ---\n")
```

```r
#   print(coeftest(models_ml[[name]]))
#
#   cat("\n---", name, "(CSS-ML) ---\n")
#   print(coeftest(models_CSSML[[name]]))
# }

for (name in names(models_css)) {
  cat("\n---", name, "(CSS) ---\n")
  print(coeftest(models_css[[name]]))

  if (!is.null(models_ml[[name]])) {
    cat("\n---", name, "(ML) ---\n")
    print(coeftest(models_ml[[name]]))
  } else {
    cat("\n---ML model missing for", name, "---\n")
  }

  if (!is.null(models_CSSML[[name]])) {
    cat("\n---", name, "(CSS-ML) ---\n")
    print(coeftest(models_CSSML[[name]]))
  } else {
    cat("\n---CSS-ML model missing for", name, "---\n")
  }
}

#========sort AIC and BIC score==========
ML_names <- names(models_ml)

sort.score(AIC(models_ml[[ML_names[1]]],
               models_ml[[ML_names[2]]],
               models_ml[[ML_names[3]]],
               models_ml[[ML_names[4]]],
               models_ml[[ML_names[5]]],
               models_ml[[ML_names[6]]],
               models_ml[[ML_names[7]]],
               models_ml[[ML_names[8]]],
               models_ml[[ML_names[9]]],
               models_CSSML[[ML_names[1]]],
               models_CSSML[[ML_names[2]]],
               models_CSSML[[ML_names[3]]],
               models_CSSML[[ML_names[4]]],
               models_CSSML[[ML_names[5]]],
               models_CSSML[[ML_names[6]]],
               models_CSSML[[ML_names[7]]],
               models_CSSML[[ML_names[8]]],
               models_CSSML[[ML_names[9]]]), score = "aic")

sort.score(BIC(models_ml[[ML_names[1]]],
               models_ml[[ML_names[2]]],
               models_ml[[ML_names[3]]],
               models_ml[[ML_names[4]]],
               models_ml[[ML_names[5]]],
               models_ml[[ML_names[6]]],
```

```r
                          models_ml[[ML_names[7]]],
                          models_ml[[ML_names[8]]],
                          models_ml[[ML_names[9]]],
                          models_CSSML[[ML_names[1]]],
                          models_CSSML[[ML_names[2]]],
                          models_CSSML[[ML_names[3]]],
                          models_CSSML[[ML_names[4]]],
                          models_CSSML[[ML_names[5]]],
                          models_CSSML[[ML_names[6]]],
                          models_CSSML[[ML_names[7]]],
                          models_CSSML[[ML_names[8]]],
                          models_CSSML[[ML_names[9]]]), score = "bic")



AIC(models_ml[[ML_names[5]]])

#============other error measures===========
names(models_ml) <- paste0(names(models_ml), "_ML")
names(models_css) <- paste0(names(models_ml), "_CSS")
names(models_CSSML) <- paste0(names(models_ml), "_CSSML")


Allmodels <- list()
fitted_models <- c(models_ml, models_css, models_CSSML)

for (name in names(fitted_models)) {
  Allmodels[[name]] <- accuracy(fitted_models[[name]])[1:7]
}

df.Allmodels <- data.frame(do.call(rbind,Allmodels))
colnames(df.Allmodels) <- c("ME", "RMSE", "MAE", "MPE", "MAPE",
                           "MASE", "ACF1")


rownames(df.Allmodels) <- c("SARIMA(8,0,8)(0,1,1)53_ML",
                            "SARIMA(7,0,7)(0,1,1)53_ML",
                            "SARIMA(0,0,4)(0,1,1)53_ML",
                            "SARIMA(1,0,4)(0,1,1)53_ML",
                            "SARIMA(0,0,5)(0,1,1)53_ML",
                            "SARIMA(1,0,5)(0,1,1)53_ML",
                            "SARIMA(0,0,7)(0,1,1)53_ML",
                            "SARIMA(4,0,4)(0,1,1)53_ML",
                            "SARIMA(7,0,4)(0,1,1)53_ML",
                            "SARIMA(8,0,8)(0,1,1)53_CSS",
                            "SARIMA(7,0,7)(0,1,1)53_CSS",
                            "SARIMA(0,0,4)(0,1,1)53_CSS",
                            "SARIMA(1,0,4)(0,1,1)53_CSS",
                            "SARIMA(0,0,5)(0,1,1)53_CSS",
                            "SARIMA(1,0,5)(0,1,1)53_CSS",
                            "SARIMA(0,0,7)(0,1,1)53_CSS",
                            "SARIMA(4,0,4)(0,1,1)53_CSS",
```

```
                                              "SARIMA(7,0,4)(0,1,1)53_CSS",
                                              "SARIMA(8,0,8)(0,1,1)53_CSSML",
                                              "SARIMA(7,0,7)(0,1,1)53_CSSML",
                                              "SARIMA(0,0,4)(0,1,1)53_CSSML",
                                              "SARIMA(1,0,4)(0,1,1)53_CSSML",
                                              "SARIMA(0,0,5)(0,1,1)53_CSSML",
                                              "SARIMA(1,0,5)(0,1,1)53_CSSML",
                                              "SARIMA(0,0,7)(0,1,1)53_CSSML",
                                              "SARIMA(4,0,4)(0,1,1)53_CSSML",
                                              "SARIMA(7,0,4)(0,1,1)53_CSSML")



df.Allmodels


#================Residual Analysis===============
residual.analysis(model = models_ml[[3]])
residual.analysis(model = models_ml[[2]])
residual.analysis(model = models_css[[9]])


#===================future measure===============
future = forecast(models_ml[[2]], h = 10)
future
plot(future)

future = forecast(models_ml[[3]], h = 10)
future
plot(future)
```