Electricity consumption forecasting using ARIMA, UCM and ML models

Gianluca Scuri[1]

[1] *University of Milano-Bicocca, Master's degree in Data Science*

February 2023

Abstract— In this project, different statistical and machine learning models are compared for the prediction of a time series. In particular, the analysed series consists of electricity consumption readings every 10 minutes and the prediction window is one month. The models tested were ARIMA, UCM and Machine Learning (SVM, RF) which, for the prediction of 4320 values (1 month), reported MAE values of: $MAE_{arima} = 1050.67$, $MAE_{ucm} = 1366.97$, $MAE_{svm} = 1649.69$. The most robust and accurate for this application turned out to be the SARIMAX model, which allowed the time series to be estimated with a relative error of 3.74%.

Keywords—Time series forecast, ARIMA, UCM, SVM, RF

1. Introduction

Time series forecasting is the use of a model to predict future values based on previously observed values. It is based on the assumption that future trends will hold similar to historical trends and has applications in many areas enabling better decision-making.

This project aim is to develop an efficient and accurate prediction model that can be used to forecast the evolution of electric consumption. In particular the model will be developed using 11 month of historical high-frequency data to generate accurate predictions one-month ahead, providing valuable insights for decision-makers in the energy sector. The time series consists of observations every 10 minutes from January to November 2017.

1. Research question

The task is to predict consumption values every 10 minutes for the month of December using statistical and machine learning techniques (4320 values). In particular, the choice of the best models between ARIMA, UCM and ML must be guided by minimising the Mean Absolute Error (MAE).

2. Data Exploration

The time series is univariate and has a regular period between every observation of 10 minutes. The data are organised in the following 2 columns:

- *date*: string encoding the date-time of the measurement, in dd/mm/yyyy HH:MM:SS format
- *power*: measured consumption

The data covers the period from 01/01/2017 00:00:00 to 30/11/2017 23:50:00 and is composed of 48096 time intervals. The *power* variable range is [13896, 52204] with a mean value of 32643 and 0 null values.

Firstly, decomposition of the time series is performed to identify the main components. As this is a time series of electricity consumption, it has prominent periods: the daily one due to the day-night cycle, the weekly one due to weekdays and holidays cycle and the yearly one due to the cycle of the seasons (not observable as the series does not contain a complete cycle). The first two periods correspond to 144 and 1008 observations as they are taken every 10 minutes. In figure 1 can be seen an example of the series while in the figure 2 can be seen its main components for January and February.

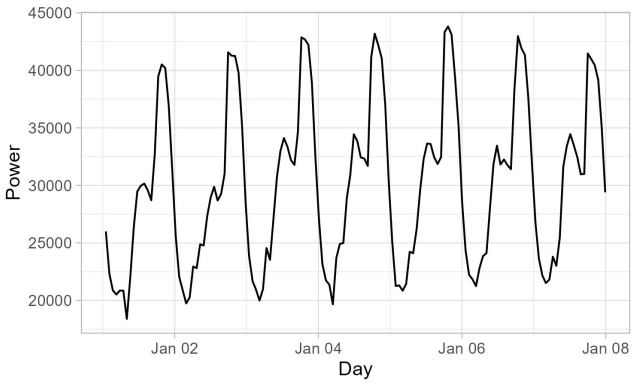

Fig. 1: Example from 1 January 2017 to 8 January 2017.

Secondly, outliers were searched for within the series, corresponding to peaks in the remainder of the complete decom-

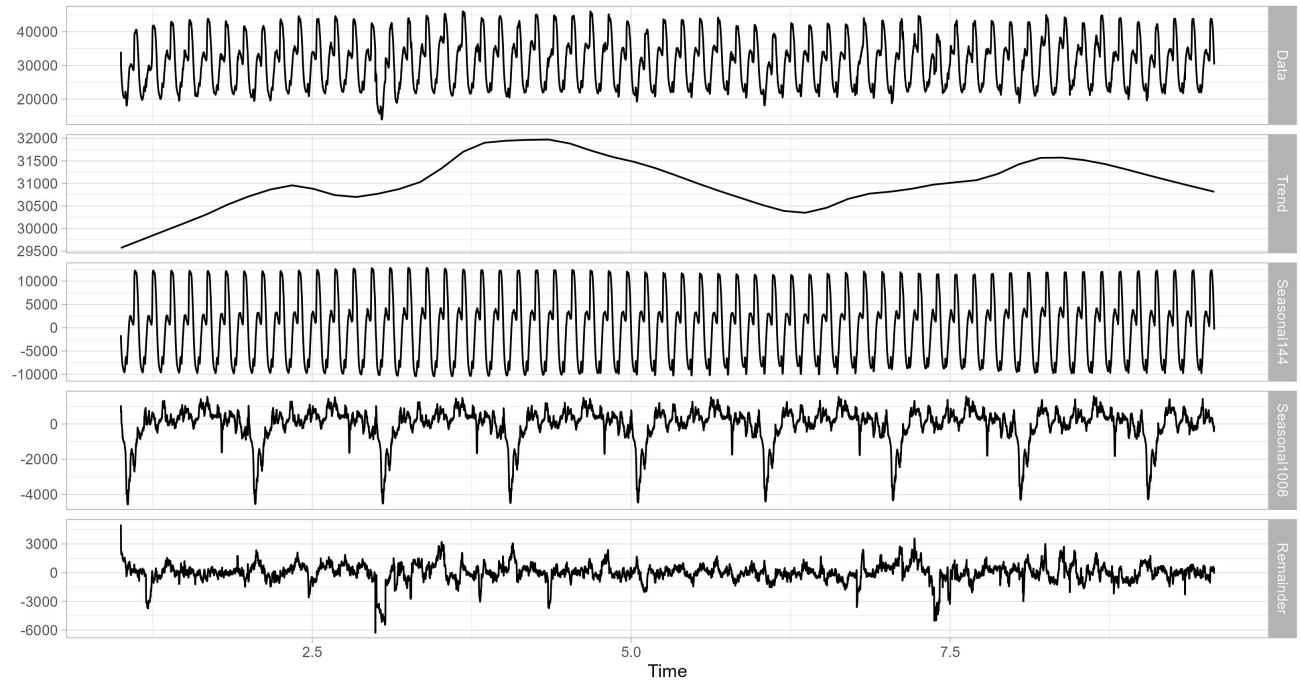

Fig. 2: Time series decomposition from January to February 2017.

position. On some days, consumption shows large fluctuations probably due to blackouts or incorrect readings (fig. 3). The problem could be solved with a ad-hoc regressor to model these outliers, but since there are only a few anomalous observations, for parsimony no action was taken.

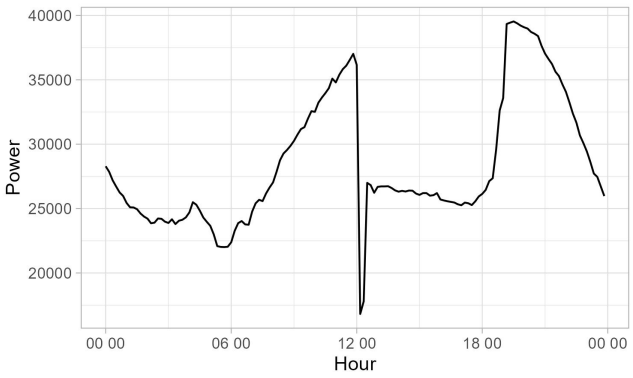

Fig. 3: Anomaly on 20th April 2017.

Then the complete series is divided into:

- Train set: from 1 Jan 2017 to 31 Oct 2017
- Validation set: from 1 Nov 2017 to 30 Nov 2017
- Test set: from 1 Dec 2017 to 30 Dec 2017 (which corresponds to the period to be forecast)

The first one is used to train the models, while the validation set is used to evaluate and compare the models.

3. Data Modelling

This section describes the modelling procedure of the time series with ARIMA, UCM and Machine Learning models. For each of these model families, different parameter combinations were tested with the aim of minimising the MAE of the predictions on the validation set.

1. ARIMA

a. Stationarity

ARMA models require the process to be stationary. As can be seen from the above graphs, the time series is non-stationary, for the following reasons:

- Presence of seasonality (daily and weekly);
- Presence of a trend (observations systematically above and below the mean);
- Possible non-stationarity in variance.

The ARIMA models, and its extensions SARIMA and SARIMAX, belong to the family of non-stationary linear stochastic processes and are an extension of the ARMA models since differentiations are applied to make the process stationary on average.

To assess whether there is non-stationarity in variance the correlation between mean and variance in one-day groups of observations is verified (fig. 4). From the graph, there seems to be a slight dependency, to fulfil the requirement of stationarity in variance the Box-Cox [1] transformations are applied.

$$x_{trasf} = \frac{x^{\lambda} - 1}{\lambda} \tag{1}$$

The automatic search function returned the value for the lambda parameter $\lambda = 0.22$.

With regard to stationarity on average, on the other hand, it is necessary to verify if there is seasonality or trend. Through the tests *ndiffs* and *nsdiffs*, which assess through the KPSS

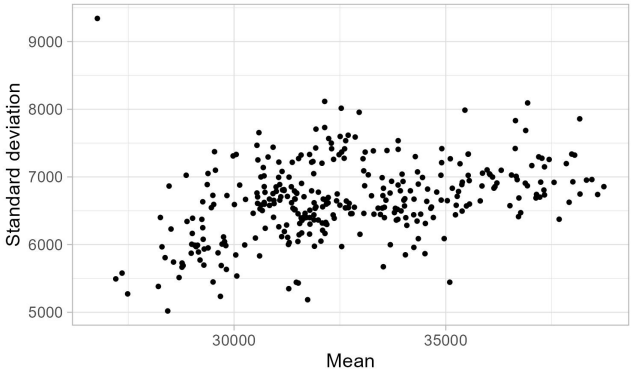

Fig. 4: Scatter plot before Box-Cox.

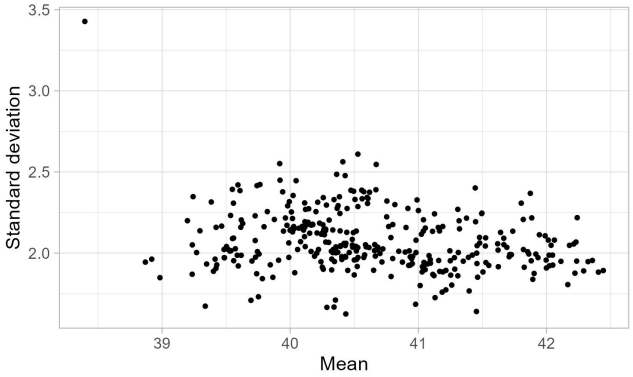

Fig. 5: Scatter plot after Box-Cox.

and Dickey-Fuller tests the need for differentiation (seasonal or non-seasonal), and through the evaluation of the Acf (fig 6) and Pacf (fig 7) graphs, it turned out that a seasonal differentiation in necessary to make the process stationary on average.

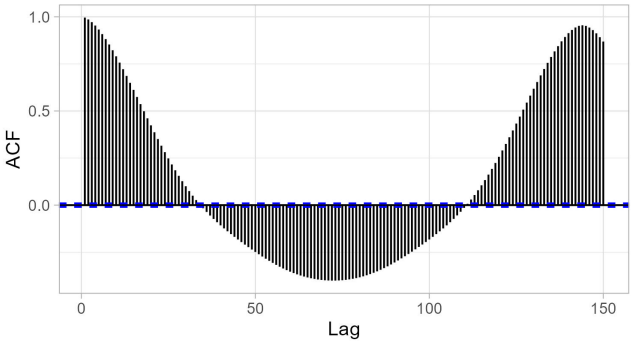

Fig. 6: AutoCorrelation Function.

The choice of parameters (p, d, q) (P, D, Q) [S] of the SARIMA model was initially made on these considerations following the Box–Jenkins method[2]. Other parameter sets were then tried and tests were performed with auto ARIMA.

b. Models

To model and predict in R the time series the function `Arima` of the package `forecast`[3] was used. The series has two seasonalities and there are several ways to model this:

- Solution 1: modelling only the daily seasonality with seasonal difference as it is the dominant one

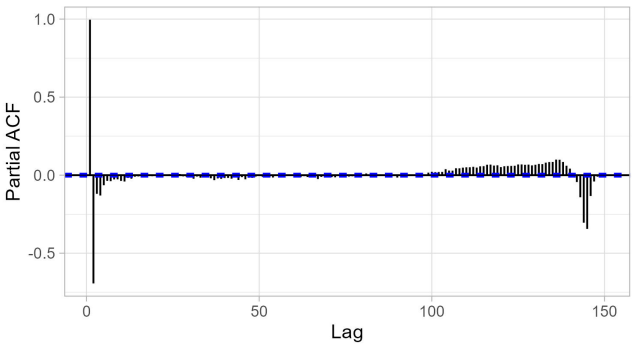

Fig. 7: Partial AutoCorrelation Function.

- Solution 2: modelling day with seasonal difference and modelling week with dummy or sinusoids
- Solution 3: 144 models to remove daily seasonality and modelling week with seasonal difference
- Solution 4: 24 models to remove daily seasonality using the hourly average and modelling week with seasonal difference
- Solution 5: hybrid solution obtained by combining the previous solutions

The first model tested corresponds to solution 1 and is based on the considerations of section a:

$$SARIMA(0,0,0)(0,1,1)[144] \tag{2}$$

This model returns an MAE value of $MAE_{train} = 1257.43$ on the training set and $MAE_{val} = 1415.86$ on the validation set. Then by iteratively evaluating the Acf and Pacf graphs on the residuals, other models were tried (e.g. eq. 3) with parameter variations, which resulted in higher MAE values. Even the search for the best model with auto.arima, with different information criteria, only proposed models with high train set description capabilities but poor generalisation capabilities (eq. 4 and 5).

$$SARIMA(0,1,0)(0,1,1)[144] \tag{3}$$

$$SARIMA(3,0,0)(0,1,0)[144] \tag{4}$$

$$SARIMA(2,0,2)(0,1,0)[144] \tag{5}$$

For solution 2, 6 dummy variables were created to model the 7 days of the week to be used as regressors. In this case, the model that has managed to generalise best is the model 6 with a MAE value of $MAE_{val} = 1050.67$.

$$SARIMAX(0,0,0)(1,0,0)[144] \tag{6}$$

Solution 3 was not analysed as measurements on such small time scales over days are highly affected by fluctuations. Instead, approach 4 was attempted by averaging 6 measurements per hour in order to have more robust values. In this case the seasonality is 7 observations and the best one-for-all model was the model 7.

$$SARIMA(0,1,1)(0,1,1)[7] \tag{7}$$

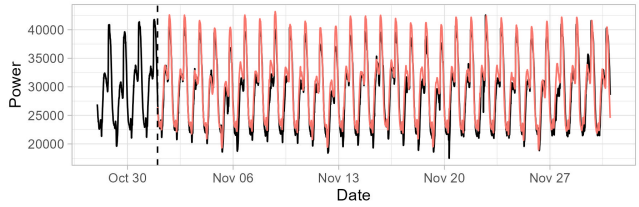

Fig. 8: ARIMA best prediction on validation set.

To regain the original series frequency, the interpolation of the values between the different hours was then made. The overall MAE value was $MAE_{val} = 1245.20$. This value is higher than that of the model 6, but, as it can be seen in the figure 9, it has much room for improvement by identifying for each of the 24 time series the best ARIMA model.

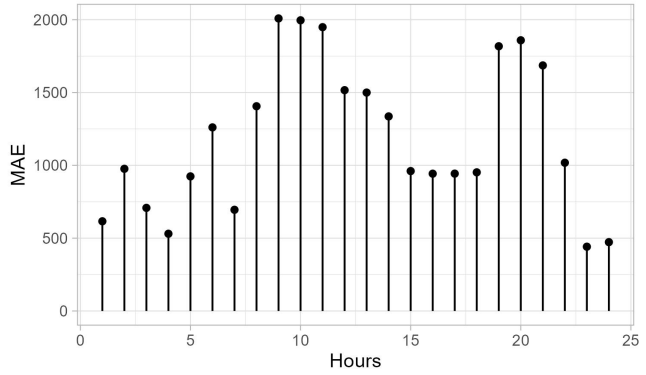

Fig. 9: MAE value per hour.

Solution 5 involves the combination of the best model on the whole series with the best model on the aggragated series. This approach allows the combination of the greatest detail of the model on the whole series with the best robustness of the model on the aggregated series. The combination can be achieved with a weighted average consisting of a single weight or 24 individual weights, one for each hour.

2. UCM

A second series of models used for the decomposition and forecasting of time series is that of Unobservable Component Models. Unlike ARIMA models, UCM models do not require the assumption of stationarity of the time series and allow the time series to be decomposed into trends, seasonality and cycles. To apply these models, it is first necessary to assign the NA value to the data to be predicted in order to allow the Kalman filter to predict values on the basis of unobserved components. Then it is necessary to examine the results of the exploratory analysis of the time series to assess the components to be included:

- Variable trend
- Strong daily seasonality
- Weekly seasonality

First models are made by modelling the time series as a Local Linear Trend and two seasonalities every 144 and 1008 observations. All UCM modelling was done in R with the `SSModel` and `KFS` functions of the `KFAS` [4] package.

$$Y_t = LLT + SEAS_{1gg} + SEAS_{7gg} \tag{8}$$

In the first configuration the seasonalities were obtained with sinusoids composed of 2 and 1 harmonics respectively. This has the advantage of being fast compared to more complex UCM model but does not allow much detail and resulted in an MAE of $MAE_{val} = 2145.91$. To increase complexity then, 10 harmonics were considered for the daily periodicity and resulted in a MAE value of $MAE_{val} = 1366.97$. Further variations were tried by changing the components (added weekly cycle) and harmonics. However, the results were not satisfying and further improvements were hampered by long train times and high sensitivity of the model to initial conditions. Tests were also made for the UCM models by creating 24 models for each time slot, but no model was found to fit all of them. Again, there is a lot of room for improvement by identifying for each hourly time series its best model.

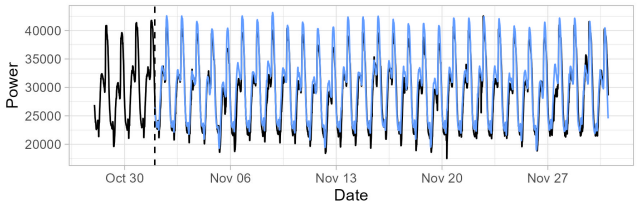

Fig. 10: UCM best prediction on validation set.

3. Machine Learning

Machine Learning models were considered as the third and last class of models. This is the most recent of the approaches to forecast time series. In the case of these models, the idea is to be able to learn the trend of the time series directly from the data, without necessarily having to manipulate it or define specific components. In this project two algorithm are used: Support Vector Machine and Random Forest. These are two of the simplest models but are capable of achieving good accuracy in many real-world cases. The temporal information contained in the data is implemented in the algorithm by passing a certain number of delays as regressors.

Regarding the SVM model the best results were obtained considering 144*7 observations as lag corresponding to the weekly periodicity. To use this model in R, the `knn_forecasting` function of the `tsfknn` [5] package was used. As Multiple-Step Ahead Strategy MIMO was used. The model returned predictions with an MAE value of $MAE = 1649.69$. Although the results are inferior to those obtained with ARIMA, this model has the advantage of being extremely fast; it proved to be the fastest among all for the prediction of future values.

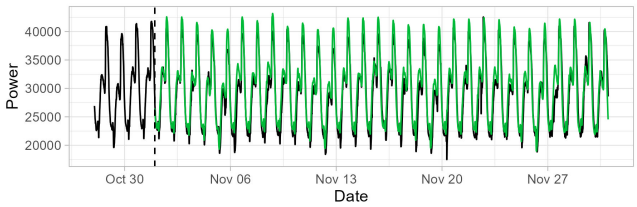

Fig. 11: SVM best prediction on validation set.

Regarding predictions obtained with RF the hourly aggregated data was used, in order to have enough lags without making it too computational heavy. A regressor was also

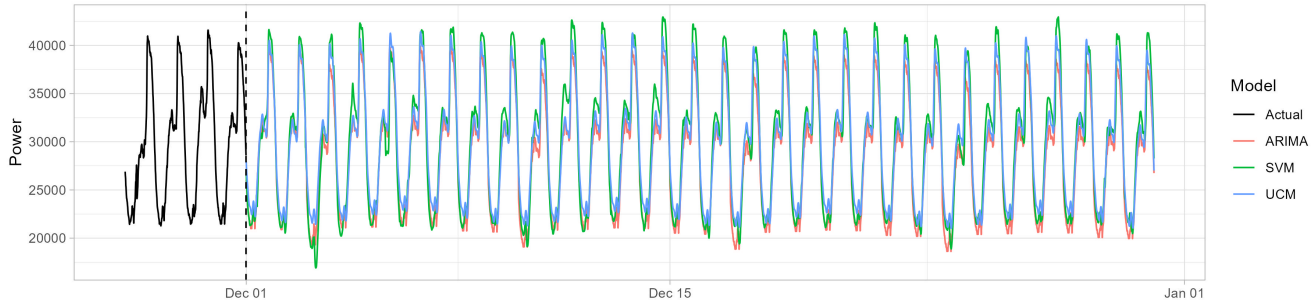

Fig. 12: ARIMA, UCM and ML comparison on December forecast.

added to indicate the day of the week. To use this model in R, the `randomForest` function of the `randomFores`[6] package was used. In this case resulted a model with a MAE value of $MAE = 1794.79$ using 200 trees and a lag of 24*7 observations (1 week). The limitations of this model are mainly two:

- the imputation of values every 10 minutes from the hourly values is quite rough;
- the variable with the day of the week is not always contained in the set of regressors used

In fact, regarding the second point, the prediction is constant over the days as it struggles to model the different days of the week.

4. Data Prediction

Here are the results of the best model for each applied technique: ARIMA, UCM, and Machine Learning.

Model	$RMSE_{val}$	$MAPE_{val}$	MAE_{val}
ARIMA	1342.44	3.74%	1050.67
UCM	1718.40	4.81%	1366.97
SVM	2082.17	6.02%	1649.69

Table 1: Results of the best models on validation set.

Each of the 3 selected models are then re-trained over the entire available time series (January-November) and predictions are made for the month of December. Comparing the results, it is easy to see that in general, the three models are in agreement with each other and provide similar predictions. It can be expected that the errors on the predicted values, barring anomalous trends, will be slightly smaller as there is an extra month of training. This is especially true for machine learning models that need a large amount of data to return good predictions. Comparing the cumulative values of the 3 predicted series results that the UCM and ML models have a similar value while the ARIMA model returns a 4% lower value.

5. Conclusion

The series under consideration being related to energy consumption is persistent and dependent on many factors. This behaviour makes it difficult to predict for all models that rely on the entire data set and not mainly on the last values, such as the ARIMA models. Analysis showed that the latter was the one that captured the most memory, leading to predictions with a relative error of 3.74%.

One approach that has potentially proved very helpful in improving all models is the aggregation of data by hour. Predictions with this system do not directly provide very accurate measurements because they lack detail, but they can be combined with the 10-minute forecasts to obtain better estimates. As future developments, one could investigate what is the best way to combine these types of prediction. In particular, the problem lies in finding the best weight or weights to be used in the linear combination. The average between predictions can also be made between the results of different models.

In conclusion, for the prediction of this time series, all the used models proved to be valid. The more traditional statistical models yielded more accurate results at the expense of long computation times due to the high number of observations in the time series. Machine Learning models, on the other hand, proved to be very functional and fast at the expense of less accurate predictions.

References

[1] G. E. Box and D. R. Cox, "An analysis of transformations," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 26, no. 2, pp. 211–243, 1964.

[2] G. E. Box and G. M. Jenkins, "Time series analysis forecasting and control," WISCONSIN UNIV MADISON DEPT OF STATISTICS, Tech. Rep., 1970.

[3] R. J. Hyndman and Y. Khandakar, "Automatic time series forecasting: the forecast package for R," *Journal of Statistical Software*, vol. 26, no. 3, pp. 1–22, 2008.

[4] J. Helske, "KFAS: Exponential family state space models in R," *Journal of Statistical Software*, vol. 78, no. 10, pp. 1–39, 2017.

[5] F. Martinez, M. P. Frias, F. Charte, and A. J. Rivera, "Time series forecasting with knn in r: the tsfknn package," *The R Journal*, vol. 11, no. 2, pp. 229–242, 2019.

[6] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 2, no. 3, pp. 18–22, 2002. [Online]. Available: https://CRAN.R-project.org/doc/Rnews/