Electricity consumption time series forecasting with ARIMA, UCM and Machine Learning models

Gianluca Cavallaro (matr. 826049)[1]

Abstract
The aim of the project is the analysis of a time series relating to energy consumption data. The time series is characterized by a 10-minute time step, starting from January to November 2017, for a total of 48,096 observations. The goal is to predict electricity consumption on the month of December 2017. The first 10 months of observations have been used to train the models, while the month of November is used as a validation set. Models belonging to the ARIMA, UCM and Machine Learning class were tested. The best results, in terms of MAE, for each class are: $MAE_{ARIMA} = 983,91$, $MAE_{UCM} = 1299,41$ and $MAE_{ML} = 1743,73$. Overall, the best solution turns out to be the ARIMA model.

Keywords
Time Series – Forecasting – ARIMA – UCM – Machine Learning

[1] *Dipartimento di Informatica, Sistemistica e Comunicazione, Università degli studi di Milano-Bicocca, Milano, Italia*

Contents

1. Introduction

Time series forecasting is the process of analyzing time series data using statistics and modeling to make predictions and inform strategic decision-making. Although it is not always an exact prediction, and the likelihood of forecast can vary greatly, time series forecasting can provide fundamental insights for evaluating the most probable future trend of the phenomenon under study. This approach has found more and more space and gained more and more importance in the digital world, especially in the industrial and economic fields. Here, a time series of electricity consumption is analysed. In particular, the goal is to compare the effectiveness of ARIMA, UCM and Machine Learning models in predicting a specific time horizon.

1.1 Research question

The task is to predict the electricity consumption values in the month of December, with a time granularity of 10 minutes. **ARIMA**, **UCM** and **ML** models are compared on the same validation set based on the Mean Absolute Error (MAE).

2. Data Exploration

As said before, the time series is **univariate** and has a regular **time-step of 10 minutes** between every observation. The dataset covers the time period between 01/01/2017 00:00:00 and 30/11/2017 23:50:00, for 48,096 total observations. The *power* variable, the one that refers to energetic consumption, is in the range [13896, 52204], without missing values. The time series looks like this:

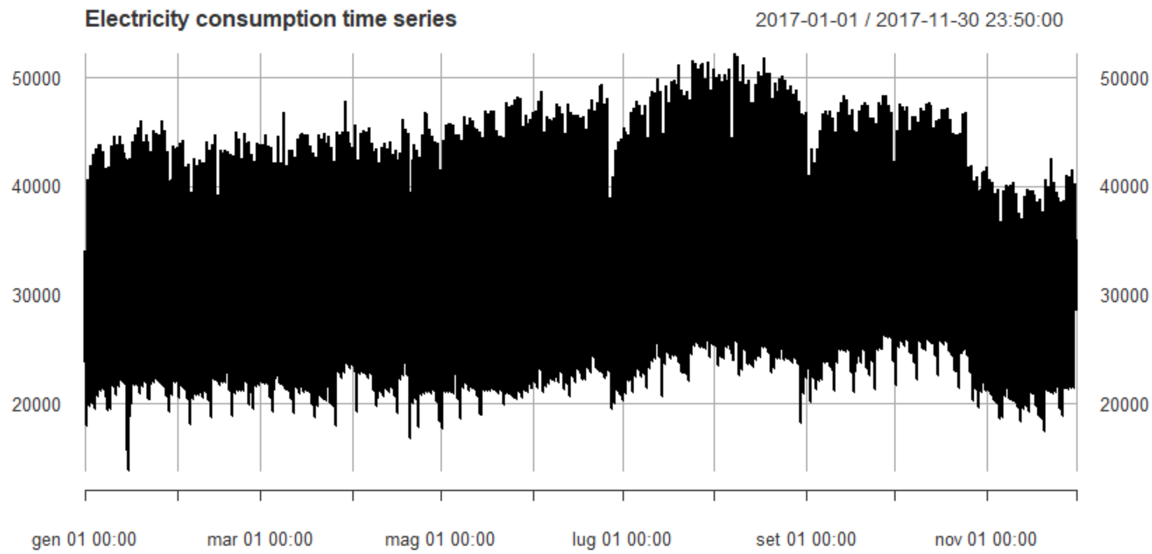

Figure 1. Original time series, 10 minutes time-step

Despite the frequency of observations, it is already possible to identify some qualitative characteristics of the series. Intra-annual seasonality can be seen, with the summer months characterized by higher energy consumption than the winter months. Here, this aspect will not be modeled specifically, having only one year of data available. The focus is more on **daily** and **weekly seasonality**, that are the two types of seasonality that predominantly characterize this type of data. These two seasonalities are shown considering only the first month of data:

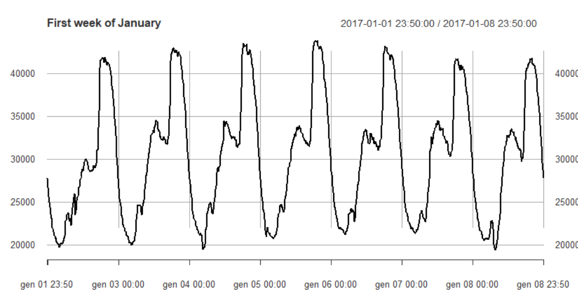

Figure 2. First week of January

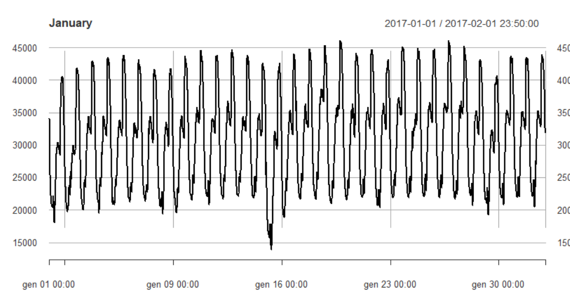

Figure 3. January

In figure 2 we can see the daily seasonality due to the day-night alternation. It can be seen that energy consumption is lowest around 1 a.m., and then increases during the day until it reaches a maximum in the evening hours. Instead, in figure 3, weekly seasonality is shown. It can be seen a kind of regular pattern: energy consumption tends to increase on weekdays, while it decreases on weekends, where most industries are closed and people spend less time at home. These two seasonalities, which in our dataset correspond to a **period of 144** and **1008 observations**, respectively, will be considered during model definition and used to construct regressors to help the models understand the time series.

We can show an overall view of these periods by exploiting the ***msts*** class of R's ***forecast*** package, designed precisely to handle time series with multiple seasonalities. To make the plot more understandable, only the first 70 days are shown:

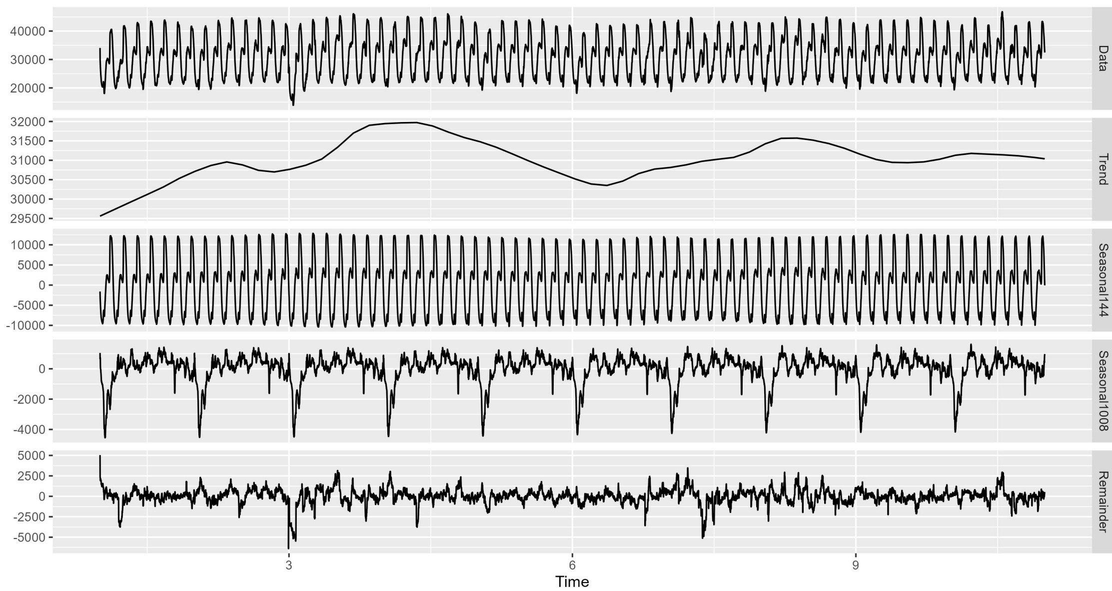

Figure 4. Time series decomposition

It can be noted that there are residuals unexplained by the two seasonalities mentioned above. Especially, a couple of negative peaks are evident. For this reason, the possible presence of outliers is assessed before proceeding with time series modeling. Considering that it is unlikely to have abrupt changes within a 10-minute interval, the difference between two consecutive observations is analyzed. A **difference greater than 5000** is arbitrarily considered to be an outlier. A number of anomalies are recorded. The most significant one is showed below:

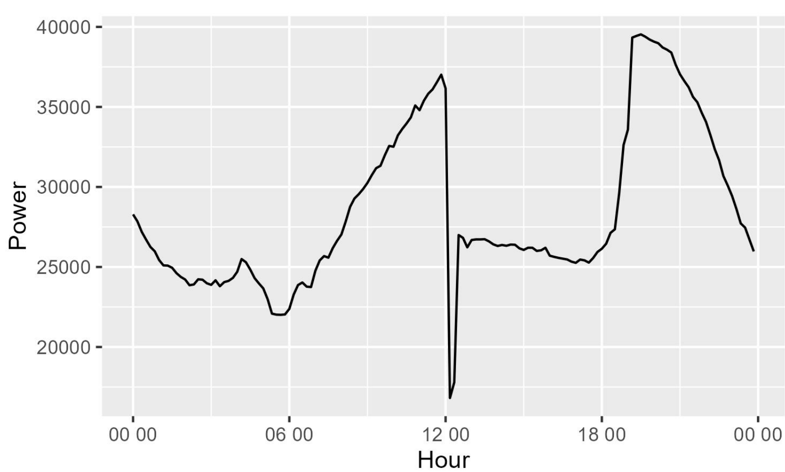

Figure 5. Anomaly registered on 20/04/2017

One might consider modeling such anomalies with a specific regressor. However, the very small number of extreme events within the dataset suggests that **no specific action should be taken**.

2.1 Training-validation-test split

The goal is to generate the forecast for the month of December. Having no way to compare the models directly on the target period, it was decided to use November as a validation set, in order to have a robust comparison of the results of the various models tested. Therefore, the dataset is divided into:

- **Training set**: from 01/01/2017 00:00:00 to 31/10/2017 23:50:00;
- **Validation set**: from 01/11/2017 00:00:00 to 30/11/2017 23:50:00;

- **Test set**: from 01/12/2017 00:00:00 to 30/12/2017 23:50:00.

3. Modelling

In this section, results obtained with ARIMA, UCM and ML models are presented. For each class of models, different sets of parameters are considered. The metric selected to determine the best model is **Mean Absolute Error (MAE)**, defined as:

$$MAE(y,\hat{y}) = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

where n is the lenght of the time series.

3.1 ARIMA

3.1.1 Stationarity

ARIMA models require the time series to be **stationary**. The current time series is **clearly nonstationary**, given the presence of a trend and multiple seasonalities. One must investigate for possible nonstationarity in mean and in variance. Starting from the latter point, a scatterplot is constructed between mean and variance, calculated by **grouping the observations by day**:

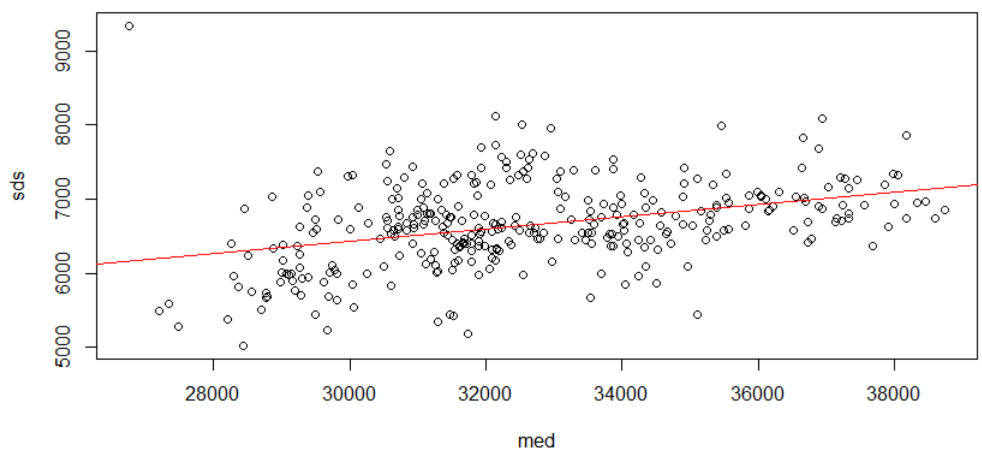

Figure 6. Mean vs. Standard deviation

Looking at the values in the graph, there is a **slight** correlation between mean and standard deviation. The optimal value of the Box-Cox transformation is calculated using the automatic search function, and it is **0.22**. However, several modeling attempts on the Box-Cox transformed time series give worse results than when the original time series is used. Therefore, it is decided **not to transform** the time series.
On the other hand, with regard to non-stationarity in mean, the Dickey-Fuller and KPSS tests reject the hypothesis of the need for a simple difference, while **emerges the need for a seasonal difference** with lag=144 to make the time series stationary.

3.1.2 ARIMA Models - Original time series

Initially, models were evaluated on the original time series[1] without the use of regressors. Starting from the assumption of the need for a seasonal difference, the first model tested is:

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

which reaches an MAE of 1415.44 on November. By looking at ACF and PACF of the residuals, we try to modify the parameters in order to eliminate the correlation. The parameters are changed iteratively, and the following models are evaluated:

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

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

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

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

[1] considering the size of the dataset, the models are trained using the CSS method which speeds up the training process.

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

$$ARIMA(3,1,0)(0,1,1)[144] \tag{7}$$

Although previous models **capture more linear memory**, with maximum ACF and PACF values of 0.06, the **MAE values** for November **are all higher than model 1**. Therefore, **adaptation to the memory of the time series results in a reduced predictive capacity**.
At this point, we look for an alternative way to model the seasonality. In particular, weekly seasonality also needs to be considered. Therefore, **regressors are introduced** to model weekly seasonality, while daily seasonality is attempted to be managed through differentiation.
First, **dummy variables** were tested. Six dummy variables were introduced in order to model the 7 days of the week. Similarly, again, the models that best collect the linear memory of the series then limited forecasting ability. Among the models tested, however, model 8 **significantly improves the MAE** over November:

$$ARIMA(0,0,0)(1,0,0)[144] \tag{8}$$

which reaches a MAE of 1065.46 on the validation set. It should be emphasized that the best model **does not use a seasonal difference**.
As a final attempt, weekly seasonality was modeled through **sinusoids**. Sinusoids with a period of 1008 were used. Several attempts were made by varying the number of sinusoids: having no great improvement as the number increased, **8 sinusoids were considered** for parsimony. Again, the same situation described above occurs. So, overall, the best model remains model 8, with a MAE on November of **1010.06**.

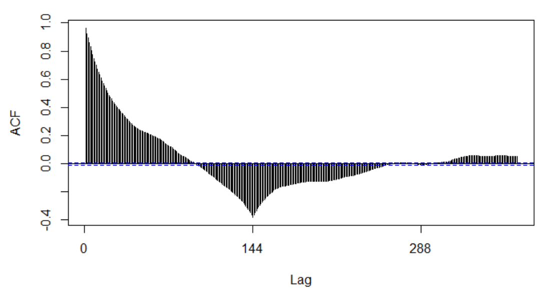

Figure 7. PACF of model 8

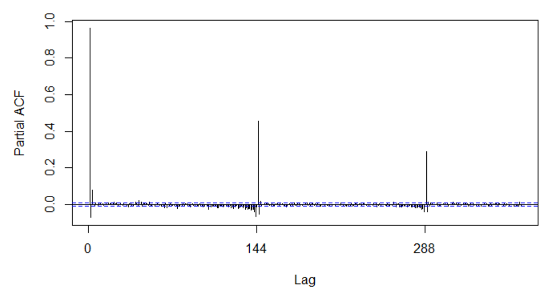

Figure 8. PACF of model 8

3.1.3 ARIMA Models - Aggregated time series

At this point, a completely different approach was tested. The time series was **aggregated by hour**, resulting in 24 different time series to be modeled independently. Taking one of the 24 series as an example, different models were evaluated, including:

$$ARIMA(0,1,0)(0,1,0)[144] \tag{9}$$

$$ARIMA(0,1,0)(0,1,1)[144] \tag{10}$$

$$ARIMA(1,1,0)(0,1,1)[144] \tag{11}$$

$$ARIMA(4,1,0)(0,1,1)[144] \tag{12}$$

$$ARIMA(0,1,1)(0,0,1)[144] \tag{13}$$

The forecasts of each of the 24 series were concatenated and the original 10-minutes granularity was restored by interpolation. The interpolation that yielded the best results was a **cubic interpolation using spline functions**. The best results were obtained with the models 12, obtained by iteratively modifying the model parameters based on ACF and PACF, and 13, suggested by the ***auto.arima*** function, who respectively obtained an MAE of 1174.46 and 1142.87. Compared with predictions obtained by modeling the original series, **ACF and PACF of the residuals** for the best models are **significantly cleaner**, as seen in the figure 7 and 8.
It should be noted that in this case the same model was used for each of the 24 time series. It is possible to compute the MAE for each hour; the result are shown in figure 11.

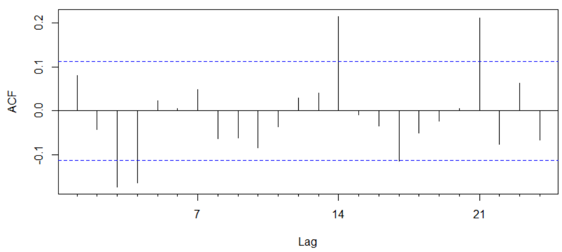

Figure 9. PACF of model 12

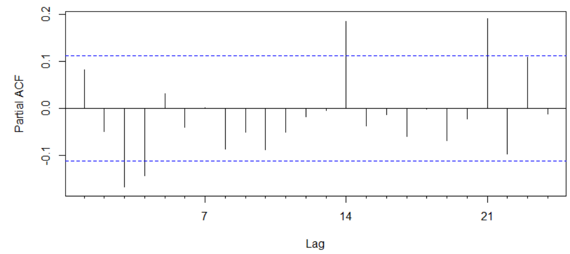

Figure 10. PACF of model 13

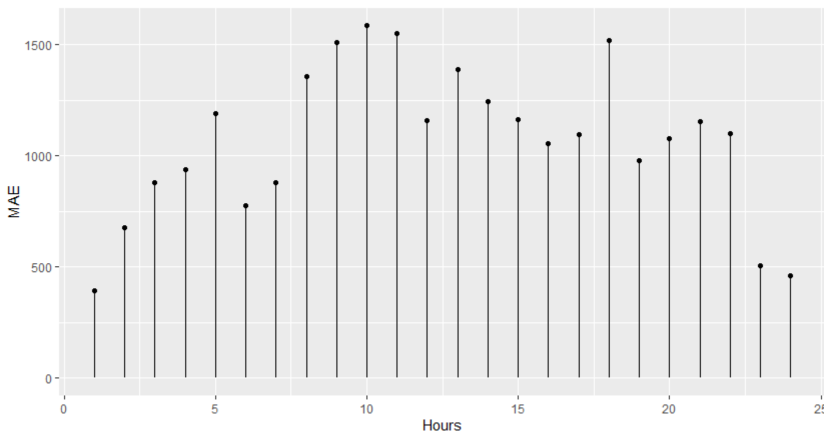

Figure 11. MAE per hour, model 13

It can be seen that the selected model leads to different results **depending on the hour**. Therefore, one possible improvement to this work is to select different parameters according to the hour.

3.1.4 ARIMA Models - Linear combination

An easy way to improve forecast accuracy is to use several different methods on the same time series, and to average the resulting forecasts. The motivation behind forecast combination is the fact that forecasting problems typically possess small or rather finite histories of points. Thus, from a practical point of view, it is not possible to obtain the correct specification of the underlying data generation process. It is beneficial to hedge against the resulting inaccuracy of the derived forecasting model by considering several forecasting models and combining their predictions.
In this case, wanting to propose separate forecasts for ARIMA, UCM and ML, of forecast combination was tested only by considering the two best ARIMA models: model 8, obtained on the original time series, and model 13, obtained on the aggregated time series. A **linear combination of the two predictions**, $\hat{y}_{orig}$ and $\hat{y}_{agg}$, was considered and the weight ω of the combination was obtained by optimizing:

$$\hat{\omega} = \text{argmin}_{\omega \in [0,1]} \; MAE(y, \; \omega \hat{y}_{orig} + (1-\omega)\hat{y}_{agg}) \tag{14}$$

where y is the real time series of the validation set. The MAE obtained with the linear combination is **983.91**, which is the best value obtained so far.

3.2 UCM

Unobserved Component Models (UCMs) are an alternative to ARIMA models, which model the time series as a sum of several components that are not directly observable, such as trend, seasonality and cycle. Each of these components is derived from deterministic functions made stochastic through the addition of random shocks.
In the current case, it is necessary to take into account the presence of a trend and the two seasonalities, daily and weekly. Therefore, the model considered is as follows:

$$\hat{y} = \text{TREND} + \text{SEAS}_{daily} + \text{SEAS}_{weekly} \tag{15}$$

Since training UCM models using the entire time series is extremely slow, it was decided to work on the aggregate series, as done in section 3.1.3. Another possibility was to train the models only on a subset of the time series, but the desire to maintain as **high generalization** as possible and the **good results obtained with ARIMA** on the aggregate series led to a preference for the first option.

Several attempts have been made:

- Initially, a model was evaluated with trend, weekly seasonality modelled trigonometrically (with only 2 sinusoids for simplicity), and by varying how daily seasonality is modeled. Using stochastic sinusoids, the estimates appear to be completely wrong, with a very high MAE. In contrast, using **stochastic dummies**, the predictions, while still leading to a rather high MAE, are more consistent.

- Next, the **optimal number of sinusoids** for modeling weekly seasonality was evaluated. Values between 1 and 16 were evaluated for the number of sinusoids: in the end, the number that led to the best result **was 6**, with an MAE of **1299.41**.

- As a final step, an attempt was made to model weekly seasonality through a stochastic cycle. A fairly good value is obtained (MAE = 1857.62), but not enough to improve on what was obtained in the previous point.

Therefore, in conclusion, the best model is one consisting of a **Local Linear Trend**, a seasonal **stochastic dummy** component (for daily seasonality) and a seasonal **stochastic sinusoid** component (for weekly seasonality). As mentioned earlier, the best result obtained on the validation set is MAE = 1299.41.

3.3 Machine Learning

The third and final class of models considered is Machine Learning models. What makes these models particularly appealing is that they can learn time-series features directly from the data during training, without having to go through preliminary steps. The temporal aspect is handled by using a number of time-series lags as regressors. It is also possible to further help the model through other regressors, as done above. The models considered in this paper are Random Forest, XGBoost and KNN. These are rather simple models, but capable of achieving satisfactory results in various real-world scenarios. All models were tested on the original time series.

Initially, the Random Forest and XGBoost models were tested using a variable number of lags, both with and without auxiliary regressors (daily and hourly dummies were used). The results obtained are not satisfactory, with the models tending to overestimate on the validation set.

The results are significantly better, although not as good as ARIMA and UCM, using the KNN model. The best result is a MAE of **1743.73**, obtained using **1008 lags** (one week of data), **MIMO** as Multi-Step Ahead Strategy, and the **median** function to aggregate the values of the nearest neighbors.

It is easily seen that the results of Machine Learning models are worse than ARIMA or UCM models, but, especially in the case of KNN, these models have the advantage of being **extremely fast in training**.

4. Summary

To summarize, the best results for each class of models are given below:

Model	**RMSE**	**MAPE**	**MAE**
ARIMA	1255.66	3.50	983.91
UCM	1666.35	4.65	1299.41
ML	2176.32	6.38	1743.73

Table 1. Results on the validation set

The comparison between the actual values of the validation set and the predictions of the three models are shown in figure 12, 13 and 14.

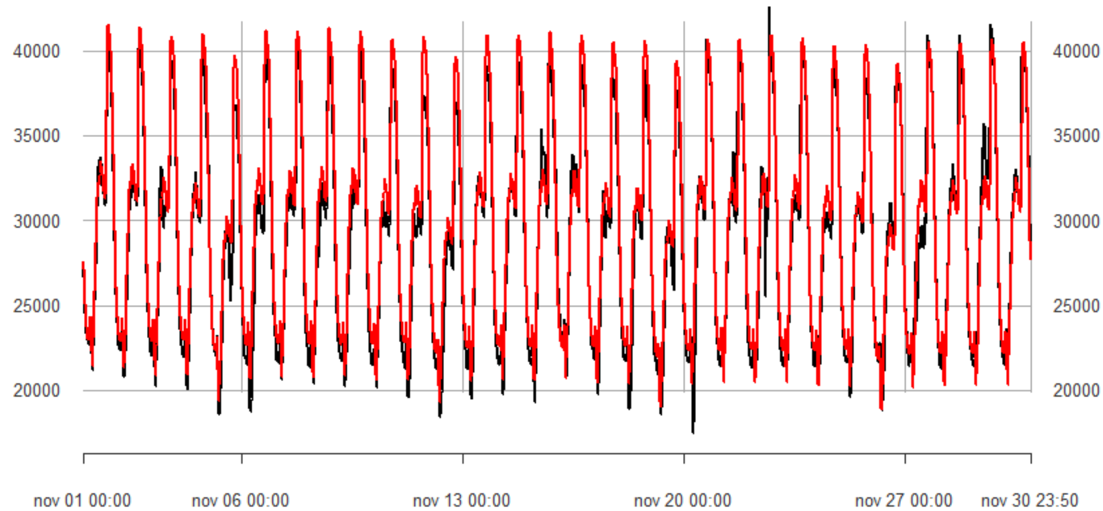

Figure 12. Validation results for the best ARIMA model

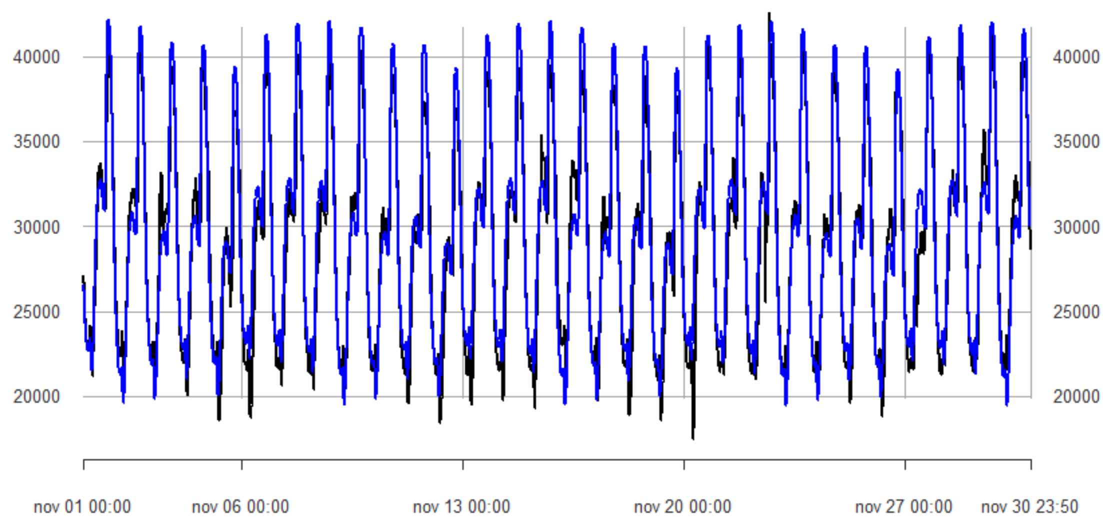

Figure 13. Validation results for the best UCM model

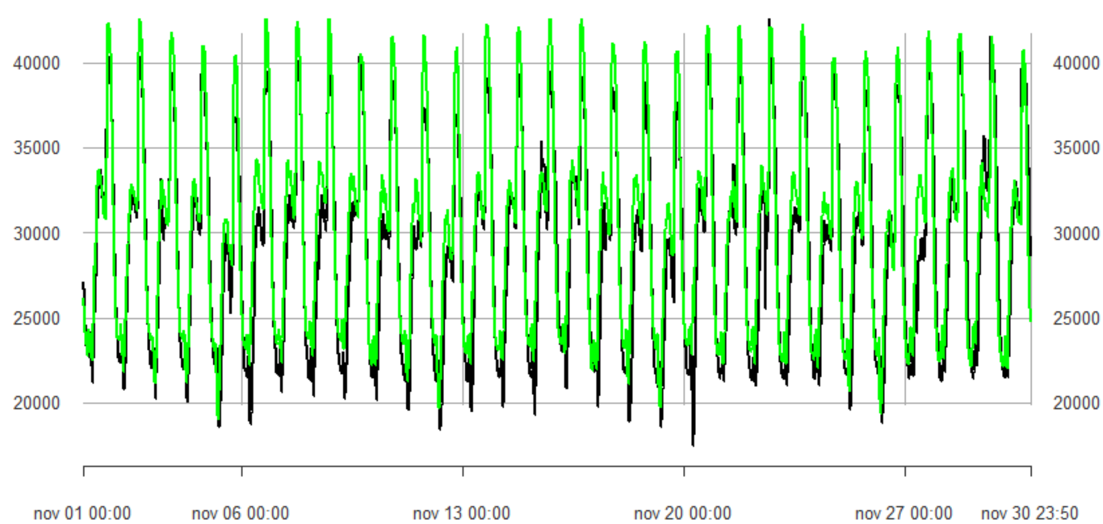

Figure 14. Validation results for the best ML model

At this point, the selected models are **retrained on the entire dataset**, including the month of November previously used as the validation set, and then forecasts are generated on December. The forecasts of the three models for December are shown in figure 15.

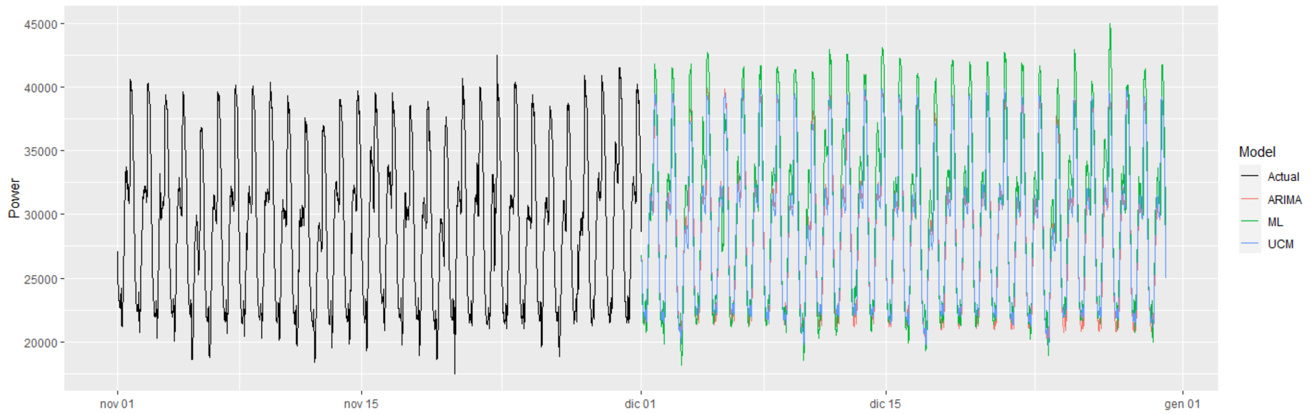

Figure 15. Validation results for the best ML model

5. Conclusions

The time series analyzed, referring to energy consumption data, is dependent on many factors. This makes the prediction task more complex for models that rely on the entire dataset, such as UCM and ML, instead of just the last values, such as the ARIMA models, which in fact led to the best results. Potentially, having **more data** and a range of **additional information**, such as holidays, weather conditions, and so on, could lead to much better results, especially in the case of Machine Learning models, which tend to perform better the more data they have available.

Despite this, the proposed approach shows that all three classes of models are valid. There is **several room for improvement**, especially in the case of ARIMA and ML models. For ARIMAs, as mentioned earlier, it would be possible to obtain better results by **modeling each of the 24 hours ad hoc**, finding different combinations of parameters that fit each of the time subseries instead of using a one-fits-all model. Instead, for Machine Learning models, it is plausible that **more complex models** than those proposed, up to and including neural networks, could lead to better results. Finally, it should be noted that, as proposed in the case of the ARIMA models, it might be useful to evaluate a **combined forecast using all three classes of models**. By finding the optimal combination, it would be possible to **exploit the strengths of each of the three strategies**.