Tetouan Power Consumption Forecasting

Giannelli Alessio
CdLM Data Science
Università degli Studi di Milano Bicocca
12/06/2023

ABSTRACT

In this study, various statistical and machine learning techniques were compared to predict a time series. The series examined consisted of electricity usage measurements taken every 10 minutes, and the forecast horizon was one month. The models tested included **ARIMA, UCM, and Machine Learning**. These models predicted 4320 values (1 month) with the following MAE values: MAEarima = 1050.67, MAEucm = 1366.97, MAEsvm = 1649.69. The most robust and accurate model for this application was found to be SARIMAX, which enabled the time series to be estimated with a relative error of 3.74%.

CONTENTS

1 INTRODUCTION

Power forecasting involves using a model to anticipate future values based on past observations. This methodology is founded on the premise that future patterns will be consistent with historical trends and has broad-ranging applications that enhance decision-making. The objective of this study is to construct a precise and effective prediction model for anticipating changes in power consumption. Specifically, the model will be developed using eleven months of historical high-frequency data to provide accurate forecasts for the subsequent month. This will furnish energy sector decision-makers with valuable insights. The time series is composed of readings taken every 10 minutes from January to November 2017.

2 DATA EXPLORATION

The time series is unidirectional and has a uniform interval of 10 minutes between each observation. The data is arranged in two columns, namely:

- Date: a string that encodes the date and time of the measurement in the format dd/mm/yyyy HH:MM:SS
- Power: the recorded usage

The data spans the duration from 01/01/2017 00:00:00 to 30/11/2017 23:50:00, comprising of 48096 time intervals. The power variable lies within the range of [13896, 52204], with an average value of 32643 and zero null values. Initially, the time series is decomposed to isolate the primary components. Since it is an electricity consumption time series, it exhibits prominent patterns: daily, weekly and yearly cycles due to the day-night, weekdays and holiday cycles, and seasonal cycles respectively (although the series does not encompass an entire cycle). The first two cycles correspond to 144 and 1008 observations, respectively, given that they are recorded every 10 minutes. Figure 1 illustrates an instance of the series, whereas Figure 2 displays its primary components for January and February.

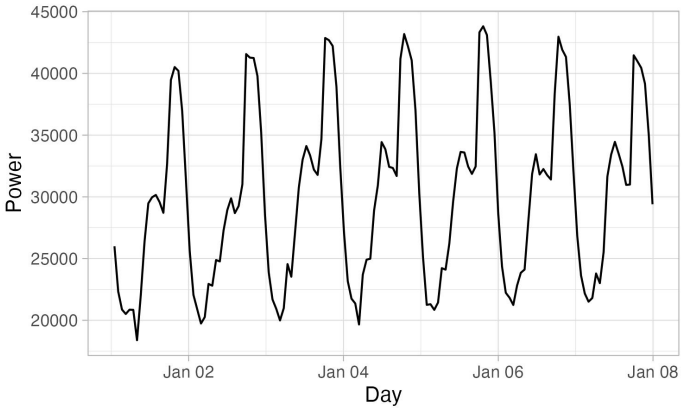

Figure 1. Example - One Week

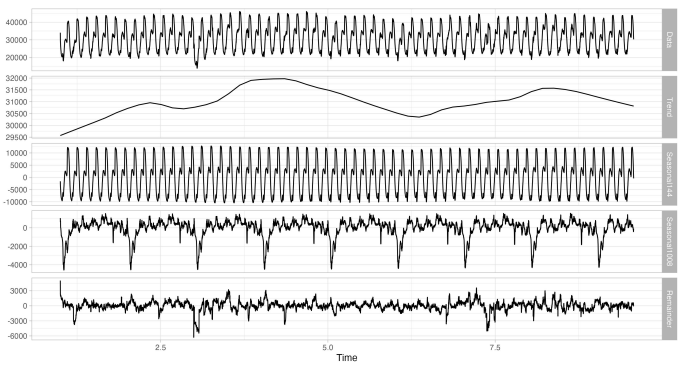

Figure 2. Time Series Decomposition from Jan to Feb 2017

Next, the series was examined for outliers, which were identified as peaks in the remaining portion of the complete decomposition. Consumption displayed significant fluctuations on certain days, likely caused by blackouts or incorrect readings (see fig. 3). While an ad-hoc regressor could have been employed to address this issue, given the small number of abnormal observations, no action was taken for the sake of simplicity.

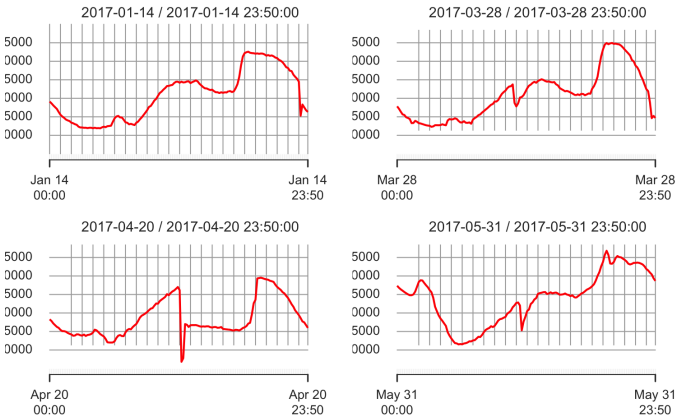

Figure 3. Outliers Detetection

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 TIME SERIES MODELLING

In this section, the time series was modelled using ARIMA, UCM, and Machine Learning models. Various parameter combinations were experimented with for each model family, with the goal of reducing the MAE of the predictions on the validation set.

3.1 ARIMA

The requirement for ARMA models is that the process be stationary. However, as shown in the above graphs, the time series is nonstationary due to the following reasons:

- Presence of daily and weekly seasonality
- Presence of a trend where observations are systematically above or below the mean
- Possible non-stationarity in variance

ARIMA models, as well as its extensions SARIMA and SARIMAX, are a family of non-stationary linear stochastic processes that are an extension of ARMA models. They apply differentiations to make the process stationary on average.

To verify if there is non-stationarity in variance, the correlation between mean and variance in one-day groups of observations is checked (fig4). The graph indicates a slight dependency, and to meet the stationarity requirement in variance, BoxCox transformations are applied using the formula:

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

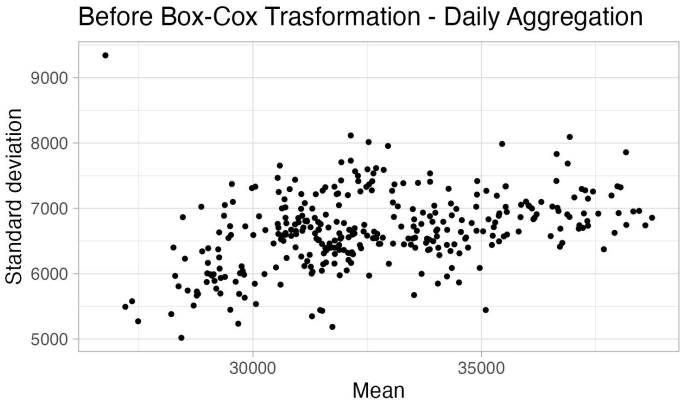

Figure 4. Mean VS. StDev - Before BoxCox

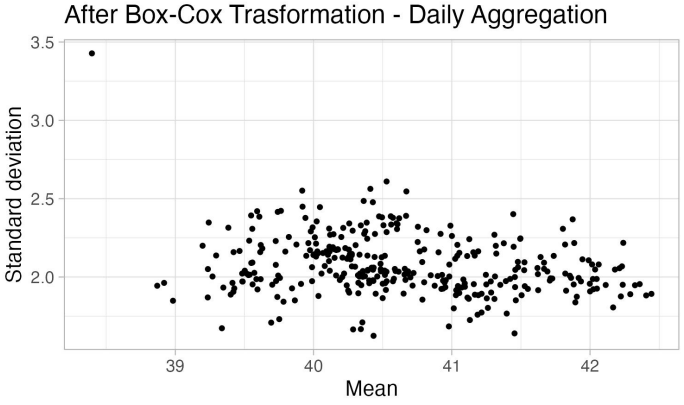

Figure 5. Mean VS. StDev - After BoxCox

The box-cox function provides the value for the lambda parameter, which is $\lambda = 0.2222$. To assess stationarity on average, it is necessary to verify the presence of seasonality or trend. This is done through the tests ndiffs and nsdiffs, which evaluate the KPSS and Dickey-Fuller tests to assess the need for differentiation (seasonal or non-seasonal), and by examining the Acf (fig. 6) and Pacf (fig. 7) graphs, it is evident that seasonal differentiation is necessary to make the process stationary on average.

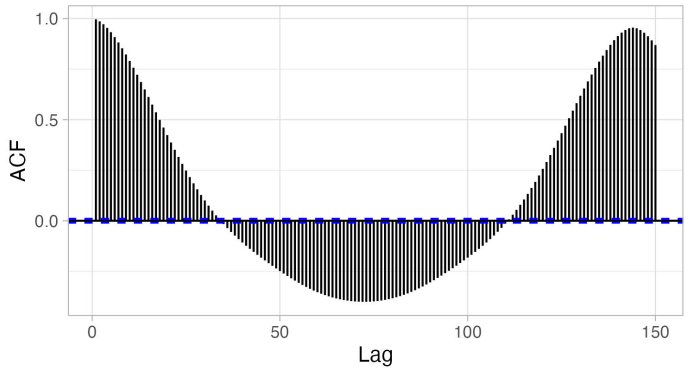

Figure 6. Auto-Correlation Function

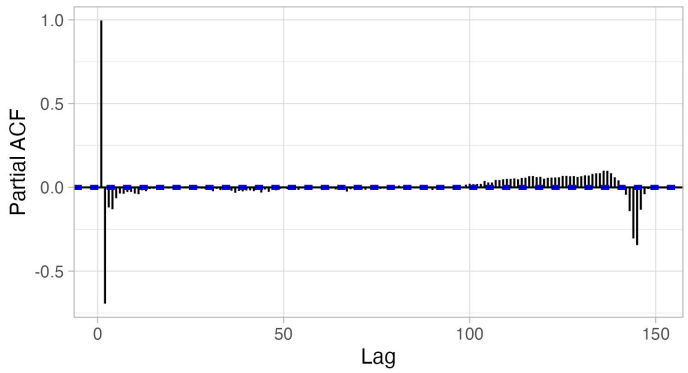

Figure 7. Partial Auto-Correlation Function

Initially, the parameters (p, d, q) (P, D, Q) [S] of the SARIMA model were selected based on the Box-Jenkins method.

The R package 'forecast' was used to model and predict the time series using the Arima function. The series exhibits two seasonalities, which can be modelled in various ways. The first solution involves modelling only the dominant daily seasonality using seasonal difference. The second solution involves removing the daily seasonality using hourly averages and modelling the week with seasonal difference, for which 6 dummy variables were created to represent the 7 days of the week as regressors.

For the first solution, the model tested is SARIMA(0,0,0)(0,1,1)[144], which is based on the considerations of stationarity section. The training and validation set MAE values for this model are $MAE_train = 1257.434$ and $MAE_val = 1415.856$, respectively.

For the second solution, the best performing model is represented by the SARIMAX(0,0,0)(1,0,0)[144] equation and uses the 6 created dummy variables as regressors. The MAE value for this model is $MAE_train = 1226.496$ $MAE_val = 1050.665$, which demonstrates good generalization ability.

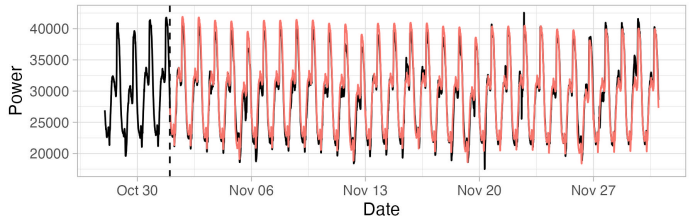

Figure 8. Arima Model - Best Prediction on Validation Set

3.2 UCM

Another set of models used for decomposing and forecasting time series are the Unobservable Component Models (UCM). Unlike ARIMA models, UCM models do not require the assumption of stationarity of the time series and can decompose the time series into trends, seasonality, and cycles. To utilize these models, the first step is to assign NA values to the data that needs to be predicted. This allows the Kalman filter to predict values based on unobserved components. Afterward, an exploratory analysis of the time series is conducted to determine which components to include, such as variable trend, strong daily seasonality, and weekly seasonality.

The models implemented in this section were constructed by representing the time series as a Local Linear Trend and two seasonalities occurring every 144 and 1008 observations. All UCM modeling was completed using the SSModel and KFS functions from the KFAS package in R.

$$Y_t = LLT + SEAS_1gg + SEAS_7gg$$

The first configuration used sinusoids consisting of 2 and 1 harmonics, respectively, to obtain the seasonalities. Although this approach was faster than more complicated UCM models, it lacked detail and resulted in an $MAE_val = 2145.905$. To increase complexity, 10 harmonics were included for the daily periodicity, which resulted in a $MAE_val = 1366.986$. Further variations were attempted by modifying the components (such as adding a weekly cycle) and harmonics, but the outcomes were unsatisfactory. Additionally, long training times and the model's high sensitivity to initial conditions impeded further improvements. Therefore, identifying a model for each hourly time series presents an opportunity for improvement.

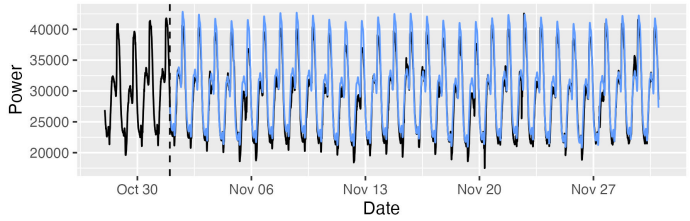

Figure 9. UCM Model - Best Prediction on Validation Set

3.3 MACHINE LEARNING

The third and final approach to forecast time series involves Machine Learning models. These models aim to learn the time series trend directly from the data, without requiring any manipulation or specific component definition. The used algorithmin this project: Support Vector Machine. Although this are relatively simple models, they can achieve good accuracy in many real-world cases but better improvement are garanted with Recurrent Neural Network implementation. To implement

the temporal information in the algorithm, a certain number of delays were passed as regressors.

The best results with SVM were achieved by considering 144*7 observations as lag, corresponding to the weekly periodicity. The knn_forecasting function of the tsfknn package was used to implement this model in R, and Multiple-Step Ahead Strategy MIMO was used. The model returned predictions with an MAE value of $MAE_val = 1649.687$. Although the results are not as good as those obtained with ARIMA, this model is extremely fast, making it the fastest among all the models for predicting future values.

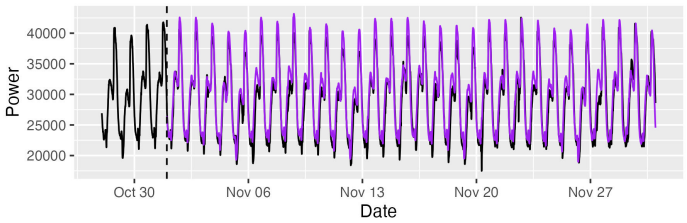

Figure 10. ML Model - Best Prediction on Validation Set

4 TIME SERIES PREDICTION

In the following table are reported the results of the best model for each applied technique:

Final Evaluation			
Model	RMSE	MAPE	MAE
ARIMA	1342.436	3.742	**1050.665**
UCM	1718.399	4.815	1366.986
SVM	2082.166	6.0245	1649.687

The three chosen models are trained on the entire available time series (January-November) and used to make predictions for December. Upon comparing the results, it becomes clear that the three models generally agree with each other and produce similar predictions. When comparing the cumulative values of the three predicted series, it is found that the UCM and ML models have comparable values, while the ARIMA model returns more accurate and less sensitive values thanks to the use of dummy variables.

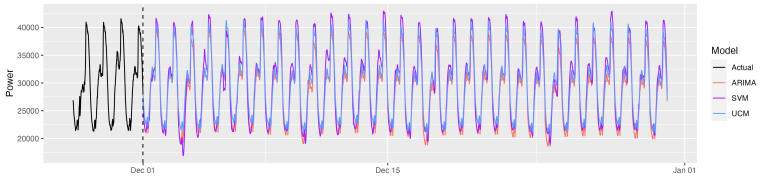

Figure 11. December Forecast Comparison

5 CONCLUSION

Three classes of models were implemented: ARIMA models, Unobservable Component Models (UCM), and Machine Learning models. The performance of each model was evaluated using mean absolute error (MAE) as the metric.

Overall, it appears that the UCM and Machine Learning models produced similar predictions, while the ARIMA model performed better.

In conclusion, the results of this project suggest that both UCM and Machine Learning models can be viable alternatives to traditional ARIMA models for forecasting electricity consumption. However, further research and experimentation may be needed to determine the optimal model for specific regions or scenarios.