ÉNERGIE ET INNOVATION : RECHERCHE / ÉTUDE & SYNTHÈSE

LES CAHIERS DE L'ÉCONOMIE

IFP SCHOOL - IFPEN N° 143 MAI • 2021

RECHERCHE

SHORT-TERM ELECTRICITY PRICE FORECASTING MODELS COMPARATIVE ANALYSIS: MACHINE LEARNING VS. ECONOMETRICS

Antoine Ferre
Guillaume de Certaines
Jérôme Cazelles
Tancrède Cohet
Arash Farnoosh
Frédéric Lantz

Pour toute information complémentaire
For any additional information

Victor Court
IFP School
Centre Economie et Management de l'Energie
Energy Economics and Management Center
victor.court@ifpen.fr
Tél +33 1 47 52 73 17

Short-term electricity price forecasting models comparative analysis: Machine Learning vs. Econometrics

Antoine Ferre[a], Guillaume de Certaines[a], Jérôme Cazelles[a], Tancrède Cohet[a], Arash Farnoosh[a] , Frédéric Lantz[a]

[a] IFP Énergies Nouvelles, IFP School, 228-232 Avenue Napoléon Bonaparte, F-92852 Rueil-Malmaison, France

[Keywords]

Energy Markets, Renewable Energy, Econometric modelling, Bootstrap Method, Merit-Order effect

(Neural Networks, Econometric Forecasting, Econometric Modelling, Bootstrap Method, Statistical Methods, Energy Markets, Energy Price Forecasting, Renewable Energy)

Abstract

This paper gives an overview of several models applied to forecast the day-ahead prices of the German electricity market between 2014 and 2015 using hourly wind and solar productions as well as load. Four econometric models were built: SARIMA, SARIMAX, Holt-Winters and Monte Carlo Markov Chain Switching Regimes. Two machine learning approaches were also studied: a Gaussian mixture classification coupled with a random forest and finally, an LSTM algorithm. The best performances were obtained using the SARIMAX and LSTM models. The SARIMAX model makes good predictions and has the advantage through its explanatory variables to better capture the price volatility. The addition of other explanatory variables could improve the prediction of the models presented. The RF exhibits good results and allows to build a confidence interval. The LSTM model provides excellent results, but the precise understanding of the functioning of this model is much more complex.

1 Introduction

The electricity spot market refers to the day-ahead market in which hourly prices for the next day are published every day around noon. Due to the fact that neither the demand nor the supply (especially from renewables) can be perfectly forecasted from a day to another, the intraday market allows to trade power contracts up to 15 minutes before physical delivery in some countries [1]. Thereafter for simplification sake, the electricity price will refer to the day-ahead price unless mentioned otherwise.

The market price of electricity is determined by the marginal cost of the most expensive plant used to produce electricity. Therefore, the addition of renewable energies such as wind and solar with almost zero marginal cost lowers the price of electricity, the so-called order-of-merit effect [2] [3] [4]. In addition, in order to make renewables increasingly competitive, many countries have introduced feed-in tariff mechanisms that allow renewable energy producers to sell their electricity at a fixed price. One consequence of these tariffs is that renewable energy producers no longer have an incentive to reduce their production when demand is low, thus bringing electricity prices down sharply. Prices can even become negative when renewables produce electricity on a feed-in-tariff basis and at the same time some power plants continue to produce electricity at a negative price because they require a minimum level of production and shutdown costs can be very high. Thus, the presence of renewable energies with chaotic production profiles (due to their intermittency) implies that prices can rise and fall at any time, leading to high price volatility.

To be able to build a robust forecasting model, electricity data have to be transparent, reliable and must come from a liquid market. The German electricity market is highly developed, and information can be easily found with an open-access database [5]. Indeed, following the Commission Regulation (EU) No 543 of June 2013, the German transmission system operators (TSO) have the obligation to supply data to the European network of TSOs (ENTSO-E). The German electricity market platform SMARD retrieves and cleans all the supplied data. Another point which makes the German market a good candidate is its growing focus on renewable energies. Indeed, Germany has a very ambitious plan for its energy transition, also called 'Energiewende' [6]. This plan gathers energy policies such as a complete nuclear phase-out by 2022, and a fully coal phase-out by 2038 [7]. In the meantime, Germany aims for renewables to account for 50% by 2030 and up to 80% in 2050. The share of renewable energies in electricity production (PV, wind and bioenergy) is growing rapidly, from 15% in 2008 to 35% in 2018 [7]. Consequently, the German electricity market is driven by renewable energies and this should be reflected in the electricity prices. Thus, building a forecasting model using German data requires to take into account the impact of intermittent energies on the grid.

Electricity prices used in that research are hourly German day-ahead prices over the years 2014 and 2015 from the EEX-Powernext platform. Solar and wind electricity generations come from the four German TSOs (TenneT, Amprion, 50Hertz, Transnet BW) at the 15 min time step (aggregated to obtain an hourly generation). Finally, the hourly electricity load was taken from the ENTSO-E database. Regarding the wind and solar data, the actual productions are used instead of the forecasts from one day to another. A consequence of this assumption would not change the nature of the models but the accuracy of the results. Moreover, adding data such as coal and gas production and prices in the case of the German market would be relevant to improve the models. In this paper, only wind and solar productions have been used.

2 Literature review

2.1 SARIMA and SARIMAX

Auto-Regressive Integrated Moving Average (ARIMA) algorithms are very effective in predicting time series, without exogenous variables. These methods introduced by Box and Jenkins in 1970s are the most comprehensive and widely known statistical methods used for time series forecasting [8]. Initially, they built a combination of autoregressive (AR) and moving average process (MA) to create an ARMA model. The AR part involves regressing the variable on its own lagged values, and the MA part involves modelling the error term as a linear combination of lagged error term. This combination is applied to a stationary time series, and the coefficients are adjusted to best fit the training series.

When a time series has a strong seasonality pattern, such as the temperature which has a daily and yearly seasonality, using a seasonal ARIMA also called SARIMA is very advantageous. SARIMA models are denoted SARIMA(p,d,q,P,D,Q)s where s refers to the number of periods in each season. The parameter p is the order (number of time lags) of the autoregressive model, q is the order of the moving average model and d refers to the degree of differencing (differencing is needed to remove the trend of the series). The parameters P, D, Q are similar but refer to the impact of the seasonality on the time series. If we denote X_t as the variable, Z_t as a white noise, Φ_p, ϕ_p , Θ_Q, θ_q as polynomial functions and B as the backshift operator, the formula of a SARIMA model is given by:

$$\Phi_p(B^S)\,\phi_p(B)\,(1-B^S)^D(1-B)^d X_t = \Theta_Q(B^S)\,\theta_q(B)\,Z_t$$

Equation 2.1.1 General formulation of a SARIMA model

The last step is the determination of the SARIMA parameters. The parsimony constraint (p + d + q + P + D + Q ≤ 6) is often respected in order to prevent the model from being too complex. As d=1 and D=1, we have the following condition: p + q + P + Q ≤ 4. The order is commonly chosen by minimizing the Akaike information criterion (AIC) or Bayesian information criterion (BIC) [9].

SARIMA model has the advantage of being easy to implement and interpret, while having a good quality of prediction. However, the algorithm fails to predict time series with high volatility and correlated with uncertain exogenous factors [10].

It is possible to integrate exogenous variables by combining a linear regression and a SARIMA process on the residuals [11]. This method is usually called SARIMAX for Seasonal Autoregressive Integrated Moving Average with Explanatory Variables. SARIMAX models are not always better than SARIMA. It depends on the quality and the relevance of input variables. In some cases, the results of SARIMAX are not better and the SARIMA models are therefore more appropriate [12].

2.2 Holt-Winters

Exponential smoothing is an empirical method of smoothing and predicting time-series data affected by hazards. However, although the moving average assigns the same weight to all past observations within a certain window, exponential smoothing results in an exponential decrease in the weight of past observations with age. The purpose of double exponential smoothing is to smooth the level of the data (i.e., to eliminate random changes) and to remove the trend. Triple exponential smoothing, or Holt-Winters method, aims to subsequently eliminate the influence of seasonality from the smoothed value, as follows:
$\{l_t\}_{t\in\{0\dots n\}}$smoothed values, $\{b_t\}_{t\in\{0\dots n\}}$ trend values, $\{s_t\}_{t\in\{0\dots n\}}$ seasonality values, (α,β,γ) tuning parameters.

$$l_t = \alpha(X_t - s_{t-m}) + (1-\alpha)(l_{t-1} + b_{t-1})$$
$$b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1}$$
$$s_t = \gamma(X_t - l_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}$$

Equation 2.2.1 Holt-Winters parameters

This method of exponential smoothing is widely used in demand projection, in order to focus mainly on the global trend [13].

2.3 Markov switching models

The literature on Markov switching (MS) models is extensive. MS models are introduced by Goldfeld and Quandt [14] and Hamilton [15]. These models can be used to predict electricity prices. For example, Haldrup et al [16] and Haldrup and Nielsen [17] studied electricity prices in the Nordic countries. They showed that these prices exhibit regime-switching and long-term memory behaviour. Other publications present MS-GARCH models with two distinct volatility regimes [18]. Other authors have studied the impact of exogenous variables on the conditional mean [19] and on the transition probabilities of inhomogeneous MS models [20] [21]. Veraart [22] models the impact of wind generation on the price of electricity. He uses a semi-stationary Lévy regime-switching process, with regimes dependent on the wind penetration index.

Markov switching models have also been used by Martin de Lagarde and Lantz [23] to make predictions on electricity prices. The idea is that prices can be either on a high price level or on a low-price level and that it is possible to switch quickly between these two regimes. This comes from the merit order effect, where prices are equal to the highest marginal cost among the means of production used (the "latest" technology used to produce electricity). If demand falls or increases slightly but involves changing the marginal technology used to produce electricity, then the resulting price will jump up or down. Price volatility is therefore very high, hence the interest of a regime-switching model.

2.4 Gaussian Mixture

The classification algorithm called the Gaussian mixture model (GMM) can be used to classify a dataset into several clusters represented by some Gaussian distributions. GMM has been applied in the biomedical field for limb motion classification for instance [24] but also to classify electricity prices [25]. An advantage of GMM over K-means is that it gives a 'shape' to the cluster with the variance of the Gaussian distribution. Two clusters with similar means would not be dissociated using a K-Means algorithm while a GMM would give more details to separate them. In the case of electricity prices that can be positive and negative and very volatile, being able to tailor the shape of the clusters is important.

GMM is a probabilistic model that assumes that data points can be considered as the realizations of a mixture of a finite number of Gaussian distributions whose parameters are unknown. To estimate the parameters of the GMM, an iterative process such as the Expectation-Maximization (EM) algorithm can be applied [26]. This method first initializes some random distribution parameters and then classifies data points by assigning the closest distribution. Then new parameters can be estimated using data points in each probabilistic cluster. The two previous steps are iteratively applied until the distribution parameters converge. The main risk of this algorithm is to overfit the data as it will converge to a situation where there are as many clusters as data points. Consequently, a maximum number of clusters must be adequately selected. For this, some statistical metrics such as the AIC or BIC can be used to find an optimal number of clusters by computing the explained variance brought by a new cluster while penalizing models with a lot of parameters.

2.5 Random forest

The random forest model (RF) has already been applied to forecast electricity prices [27] [28]. An interesting feature that makes this algorithm suited for this task is the probability distribution returned by the RF [27], which allows to evaluate the accuracy of the prediction. RF has also been used to predict some clusters of prices such as low, medium and high ranges [29].

A RF is composed of several decision trees. Each of them performs classification tasks using the so-called CART algorithm [30]. The main feature of a random forest is the bootstrapping, i.e the generation of random groups of samples to train a decision tree at each stage of the bootstrapping. This repetition of training strongly mitigates the risk of overfitting. At each stage, some samples are left aside and are not picked to train the decision tree, thus they can be used to evaluate an error. This process is called the out-of-bag (OOB) error. This is equivalent to a cross-validation metric with the advantage that it is being measured during the training stage and therefore it is computationally cheap. The OOB error can be used to tailor the parameters of the random forest such as the number of trees or the depth of the decision trees.

2.6 Neural Networks

Neural networks have been used since the 1950s and have been continuously improved [31] .However, neural network computations require a large database in order to train the network correctly, as well as high computing power from the computers. These models therefore became obsolete in the 1970s. A renewed interest in neural networks appeared in 1990, thanks to a very large flow of data, more powerful computers, and the development of new, more complex models through supervised learning. Numerous publications have been written on neural networks [32].

Recurrent neural networks (RNNs) are used to process sequences. This applies very well to time series. Several layers of neural networks follow one another, and the algorithm drives these networks using the gradient descent algorithm. The big disadvantage of this model is that when processing large amounts of data, the "vanishing gradient problem" can occur [33] [34]. As a reminder, the vanishing gradient problem is the fact that the sigmoid function that makes the prediction takes very small or very high values, but the gradient associated with those extreme values will be very small and thus, the weights are not modified correctly at each step. It means that if the previous layer has not a good value, it will not be corrected for the next one and so on. Finally, the prediction will not be very good.

Long short-term memory (LSTM) is an artificial recurrent neural network architecture used in the field of deep learning. It can not only process single data points (such as images), but also entire sequences of data (such as speech or video). Its main advantage is that for a big database it does not have the vanishing gradient problem when the classic RNN do.

This model was proposed by Hochreiter and Schmidhuber in 1997 [35]. This model uses 3 different gates. What are called "gates'' are in fact neural networks that make it possible to regulate the flow of information that circulates in the sequence chain.

3 gates are used in this algorithm:

- ***the forget gate:*** used to free the memory from “unnecessary” data
- ***the input gate:*** used to add new data

- ***the output gate***: from the information it forgot, from the new data and knowing the data it used to stock before, it determines the new state of the cell at time t.

3 Methodology

3.1 Presentation of the dataset

In order to build price forecasting models, the first step is to visualize the dataset. As we could expect it, we observe prices mainly around 25-55€ with high and very low picks. Those variations in prices might be explained by several factors: oversupply of electricity coupled with a lack of demand and excess of demand coupled with a low production. To look whether it exists a seasonal trend, we then look at a smaller scale in Figure 3.1.1.

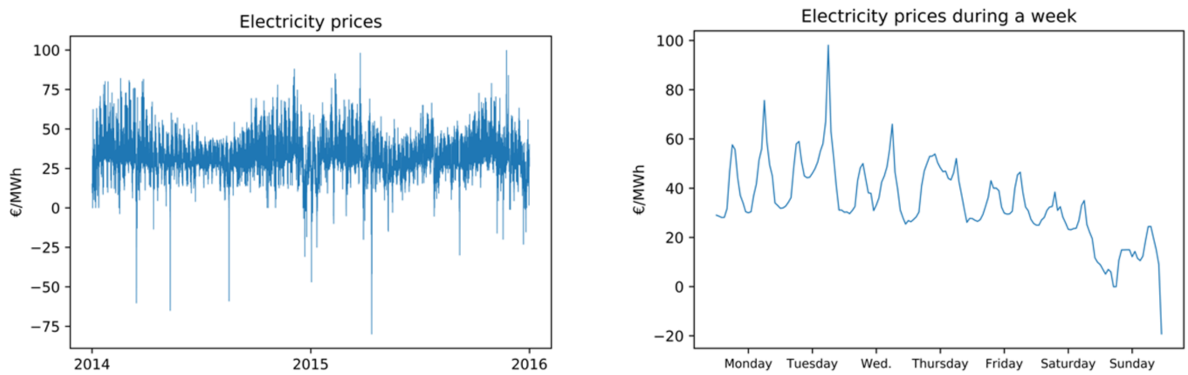

Figure 3.1.1 German day-ahead electricity prices over a 2-year period and in March 2015

We clearly notice a seasonal trend that we will try to determine more precisely with some statistical tests. It seems that the seasonality is 24 hours (we notice the same daily pattern).

3.2 Transformation function

It is clear that electricity prices are not a stationary process. However, it is possible to make this dataset stationary by applying a transformation function to it. The logarithmic transformation function is often found in the literature and has the advantage of reducing the weight of extreme values and reducing non-normality. Nevertheless, this function is not appropriate in our case because we have negative data (up to -79.9€/MWh). As suggested in [36], the solution could be to translate all the data by 80€/MWh but this would result in compressing the majority of the values while giving too much importance to the extreme negative values. We therefore chose to use the inverse hyperbolic sinusoidal function in Equation 3.2.1, that was originally described in [37]. More recently, it has been used by Schneider in [38] as well as by Martin de Lagarde and Lantz in [23] for electricity prices.

$$f(x, \xi, \lambda) = \sinh^{-1}((x - \xi)/\lambda)$$

Equation 3.2.1

This transformation function has the advantage of compressing extreme values while spreading out intermediate values. There are many ways to choose ξ and λ but we decided to take ξ as the mean of our data set and λ as 1 €/MWh (for simplicity purposes).

After transformation, the resulting data illustrated in Figure 3.1.1. The transformed data have no extreme values and are centred around 0.

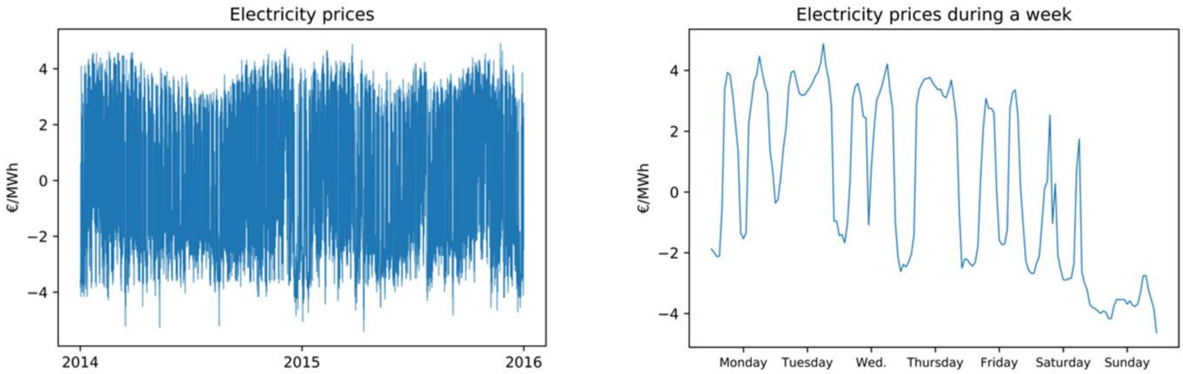

Figure 3.2.1 German day-ahead electricity prices over a 2-year period and a week in March 2015 after transformation

3.3 Econometric approach

3.3.1 SARIMA model

Once the transformation function has been applied to our dataset, the next step is to study the autocorrelation function (ACF) and partial autocorrelation function (PACF) applied on the electricity prices to check whether there are some correlations within our dataset. In the following, an analysis of the ACF and the PACF has been conducted and the results are plotted in the Figure 3.3.1.

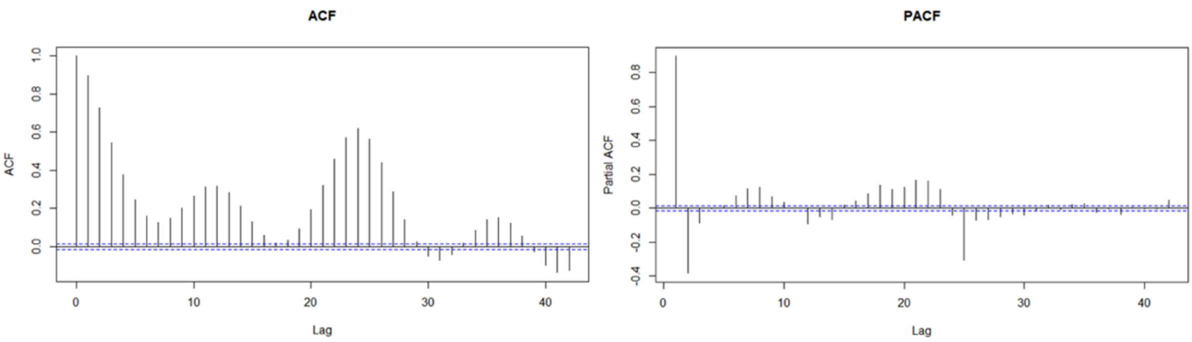

Figure 3.3.1 ACF and PACF of the transformed electricity prices

Autocorrelation is present in the dataset since there are a lot of spikes in the ACF graph. The big spike at 24 in the PACF and in ACF suggest a seasonality of 24 hours, as one might expect. However, it is hard to determine with the graphs the parameters p, q, d, P, Q, D. For the moment we can only guess that there will surely be at least a one-order of non-seasonal differencing and a one-order of seasonal differencing.

The objective is to find the parameters for the SARIMA(p, d, q, P, D, Q)$_s$ model that will enable the model to provide the best predictions. We would like to respect the parsimony rule p + d + q + P + D + Q ≤ 6. We therefore test several models respecting this condition, and we choose the one that minimizes the AIC crystals.

After trying many different models, we decided to keep the SARIMA(0, 1, 2, 0, 1, 2)$_{24}$ model since it has the lowest AIC value. If we define B as the backshift operator such as $B\,X_t = X_{t-1}$, the mathematical notation of the model in Equation 3.3.1 and Equation 3.3.2. We note Z_t the white noise and X_t the transformed price at period t.

$$(1-B^{24})(1-B)X_t = (1+\theta_1 B+\theta_2 B^2)(1+\Theta_1 B^{24}+\Theta_2 B^{48})Z_t$$

Equation 3.3.1

$$\Leftrightarrow X_t = X_{t-1}+X_{t-24}-X_{t-25}+Z_t+\theta_1 Z_{t-1}+\theta_2 Z_{t-2}+\Theta_1 Z_{(t-24)}+\theta_1\Theta_1 Z_{t-25}+\theta_2\Theta_1 Z_{t-26} + \Theta_2 Z_{t-48}+\theta_1\Theta_2 Z_{t-49}+\theta_2\Theta_2 Z_{t-50}$$

Equation 3.3.2

In which the coefficients θ1, θ2, Θ1, and Θ2 are given in Table 3.1:

$ma1\ (\theta_1)$	$ma2\ (\theta_2)$	$sma1\ (\Theta_1)$	sma2 (Θ_2)
0.0797	-0.0485	-0.757	-0.132

Table 3.1 Coefficients of the SARIMA model returned by the software R

The residue analysis is shown in Figure 3.3.2. It can be seen that the autocorrelation in the residuals is almost zero. The residuals are not perfectly aligned on a straight line due to extreme values: the assumption of a normal distribution of the residuals is not perfectly respected.

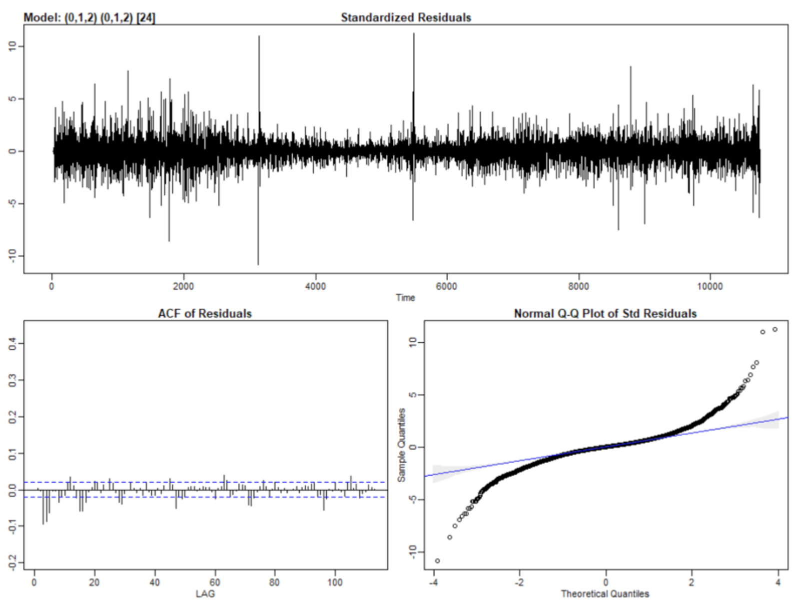

Figure 3.3.2 Analysis of SARIMA residues

3.3.2 Linear regression coupled with an autoregressive approach on residues

Electricity prices are closely linked to exogenous variables such as wind and solar generation and plant load. SARIMA's problem is that it does not take these variables into account. Linear regression is interesting because we have variables relevant to our model. The significance of the different variables was tested with the anova test in Table 3.1. We kept the significant variables which maximized the R^2. The chosen regression is illustrated in Equation 3.3.3.

$$X_t = \mu + \gamma_1 Wind_t + \gamma_2 Solar_t + \gamma_3 Load_t + \sum_{i=1}^{6} \alpha_i\, Day_i + \sum_{i=1}^{11} \beta_i\, Month_i + \epsilon_t$$

Equation 3.3.3

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Wind	1	359 912	11 9970	1 537	< 2.2e-16
Solar	1	97 170	32 390	415	< 2.2e-16
Volume	1	17 5817	58 605	751	< 2.2e-16
Day	6	243 448	40 574	520	< 2.2e-16
Month	11	24 814	2 255	28	< 2.2e-16
Residuals	10747	838 318	78.00	NA	NA

Table 3.2 Anova test of linear regression variables

Once the regression is applied, using a Ljung-Box test will allow to test whether autocorrelations exist. The ACF and PACF diagrams show that there are still autocorrelations in the residues Figure 3.3.3 and the residues have been plotted in the Figure 3.3.4:

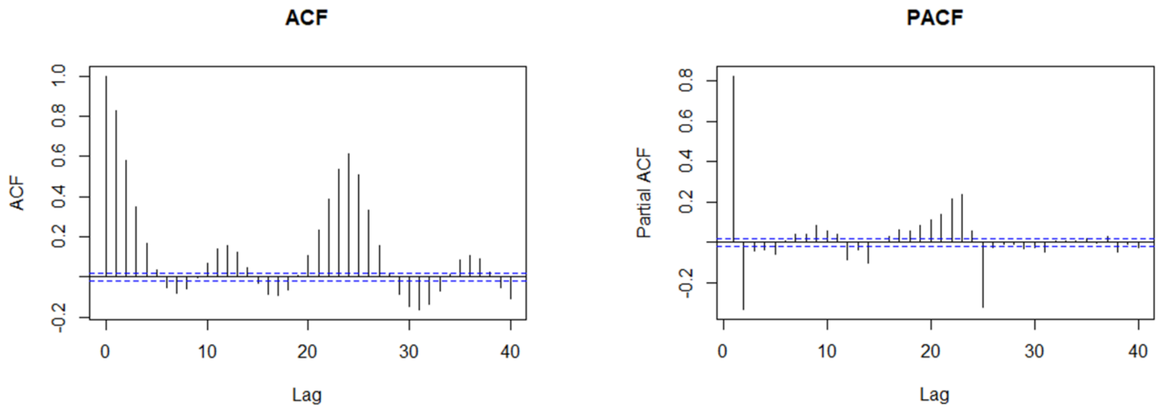

Figure 3.3.3 ACF and PACF of the transformed residues

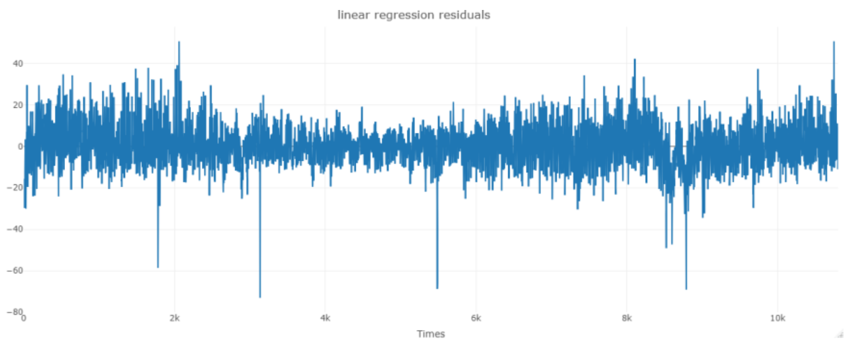

Figure 3.3.4 Plot of the residues of the linear regression

The residues do not follow a stationary process, so we applied the same transformation function except that we used a higher value of lambda (10 instead of 1). We then, found the best SARIMA model that fits the residues (in the sense of the AIC criterion), which happens to be the same than the SARIMA model presented previously. The final model is obtained by aggregating the linear regression and the SARIMA forecast.

3.3.3 Monte Carlo Markov Chain Switching Regimes Model

Instead of modelling a regime change with Markov processes, it is also possible to parameterise this with random forest. This has the advantage of being simpler to build with high performance results. The idea is therefore to separate the data set into 2 price regimes (low price and high price) using the transformation function. As shown by Martin de Lagarde and Lantz [23], these 2 regimes obey very different economic laws, with a demand-supply structure that is not the same, due to the intermittency of renewable energies. Therefore, it is interesting to apply a linear regression for each of the 2 regimes in order to represent the structural difference of the two situations.

The separation of prices into 2 regimes can be done simply with the average of prices (regime 0 for prices below the average and regime 1 for the rest). Afterwards, we are making regression on the Renewable and Pilotable Load features, plus some additional qualitative features (day, month, hour of the day, last price level the day ahead), with Gradient Boosting Regressor on each regime.

After having assigned a regime to each time step, we are training a Random Forest Classifier on the previous features in order to forecast both price regime and to apply the right regression. Therefore, the model does 2 steps: it first chooses the price regime (0 or 1) in which the dataset is most likely to be, then it predicts the price with the associated regression. In this way, extreme values can be well predicted but we will have to check that the model is indeed continuous for intermediate prices.

3.4 Machine learning approaches

3.4.1 Forecasting electricity prices with a random forest algorithm

The goal of this section is to first classify the electricity prices into several clusters using a GMM. Then, a RF is applied to forecast the clusters associated with each price of the next 24 hours.

The electricity prices have been transformed using the inverse hyperbolic sine function described in Equation 3.2.1. As discussed in the previous section ξ is set equal to the mean of the prices and λ is 1. The classification has been applied on the electricity transformed prices using the GMM from the Scikit-learn package on Python [39] [40]. The optimal number of clusters was determined using the AIC and BIC metrics and the results are shown in Figure 3.4.1 for up to 50 clusters. The first 10 clusters play a major role in increasing the accuracy of the model. However, while the AIC keeps decreasing for a higher number of clusters the BIC criterion tends to increase slowly around 30. Consequently, in the following, all the results will be based on a classification using 30 clusters.

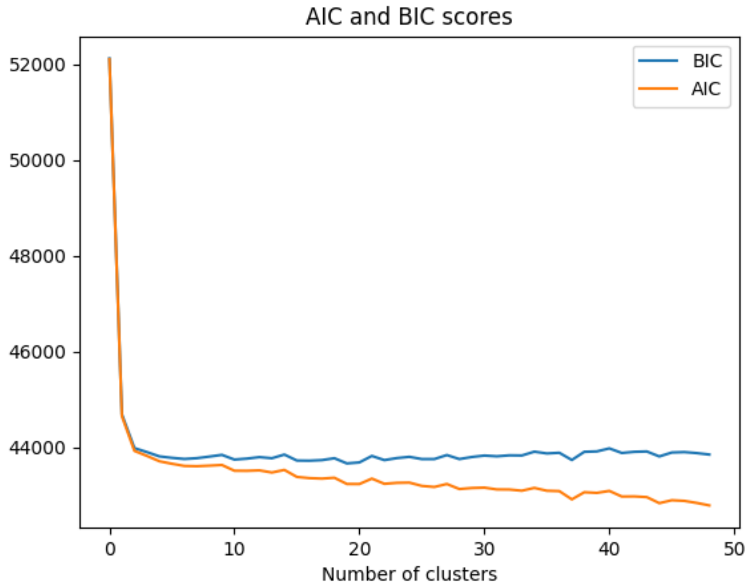

Figure 3.4.1 AIC and BIC metrics of the GMM classification

The GMM has been trained for this number of clusters on the transformed prices. A random sample of the same length as the price data has been computed using the trained GMM and compared to the histogram of the real transformed prices. Both graphs are illustrated in Figure 3.4.2.

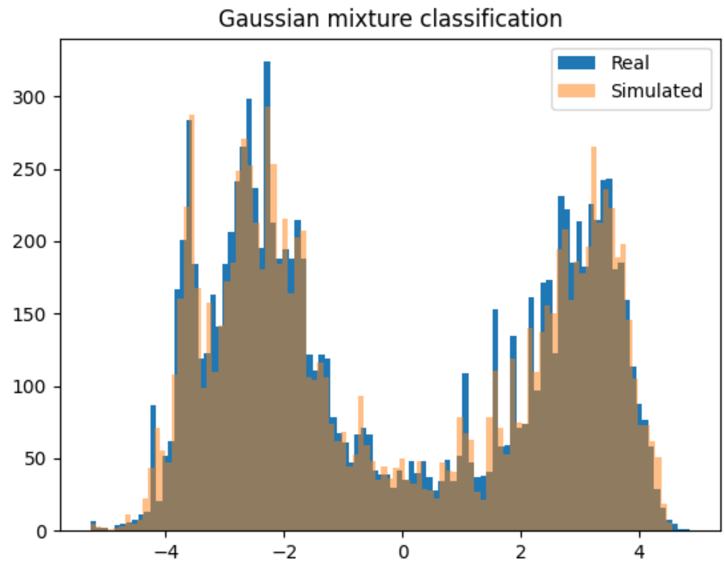

Figure 3.4.2 GMM classification against a histogram of real prices

The GMM seems to capture several price distributions of different shapes. Of course, two main regimes (low and high prices) can be well separated and that results in the large drop in the AIC and BIC scores in Figure 3.4.1. However, adding more clusters (up to 30) allows us to dissociate smaller price distributions while improving the accuracy of the forecasts without overfitting our model.

Once the prices have been classified, it is possible to apply supervised algorithms to forecast the clusters, i.e the range in which the future prices should be found. The aim is to map a vector of inputs to a cluster which is represented by a Gaussian distribution with known parameters. First, a vector of inputs for each data point must be determined. Obviously, this vector must not take into account the price of the time that we wish to forecast (expressed in hours) but rather the price of previous dates. As mentioned before, we would like to forecast the 24 prices of the next day, consequently only the prices prior to t-24 should be considered. The presence of autocorrelations in electricity prices and some seasonal effects (24 hours cycles) should be taken into account in the vector of inputs. Additionally, the information on the date (month, week, day and hour) can also provide interesting features to the model. Finally, the solar and wind production of the day were also considered.

The vector of inputs for one data point is shown in Table 3.3

Month	Day of the week	Solar production (next 24h)
Week	Price of the previous day	Wind production (next 24h)
Day of the month	Price of the previous week	Volume (next 24h)
Hour	Price of the previous month	

Table 3.3 Vector of inputs

Finally, the array of inputs is a matrix in which the rows correspond to one data point, i.e one hour of a specific date and the columns correspond to the variables defined in the previous table. Now that the inputs have been computed, a vector of outputs representing the labels of the clusters associated to each data point must be determined. For this, it only requires to read the outputs of the GMM for each hour. Lastly, the resulting dataset (inputs and outputs) has been cut into a training set from the 1st of January 2014 to the 24th of March 2015 and a testing set from 25th to 27th of March 2015.

The Random Forest (RF) has been trained on the data using the matrix of inputs and vectors of outputs previously introduced, 1000 decision trees and a maximum depth of 30 have been used. The random forest is returning probabilities which can then be used to build confidence intervals.

We can evaluate the quality of the prediction with the confidence interval. The tightest the interval is, the more confidence we will have in the prediction. We predicted the prices with the expected values, based on the probabilities relative to each cluster. We can also calculate the variance of the clusters, which is given in Equation 3.4.1, for a discrete random distribution, for N clusters and with $x_0...x_N$ the means of each cluster.

$$\mathrm{Var}(\mathrm{X_t}) = \sum_{i=1}^{N} p_i(t)\big(x_i - E(X_i)\big)^2$$

Equation 3.4.1

Where X_t is a random variable which takes its values in the set composed of the cluster means {x0...xN }. $p_i(t)$ is defined as $p_i(t) = p_i(X_t = x_i)$ which is the probability that the variable X_t belongs to the cluster i of mean x_i. We have a variance for each time of the prediction. The variance shows the certitude of the classification. For example, if there is no distinct probability higher than the others, the variance will be significant. On the contrary, if a probability is around 1, the variance will be very low. Then, we can find a confidence interval with the Bienaymé-Thebychev inequality. We thus have the confidence interval at the α threshold and for a standard error σ, according to the Bienaymé-Tchebychev inequality given in Equation 3.4.2.

$$P(|X\_t - E(X\_t\ )| \geq \alpha) \leq \sigma\hat{}2/\alpha$$

Equation 3.4.2

At the 95% threshold, the confidence interval is therefore given by Equation 3.4.3

$$]E(X_t) - 4.45\ \sigma;\ E(X_t) + 4.45\ \sigma[$$

Equation 3.4.3

The RF returns a probability distribution for each hour of the testing set. Confidence intervals using the Bienaymé-Thebychev inequality are wide because the inequality is true for all distributions. However, we are sure that this interval is valid for the prediction of the clusters. We can have a tighter prediction if we approximate the probabilities of clusters as a normal distribution. Although, this approximation is not valid for extreme cluster, we have assumed that the probability distribution illustrated in Figure 3.4.3 is normal:

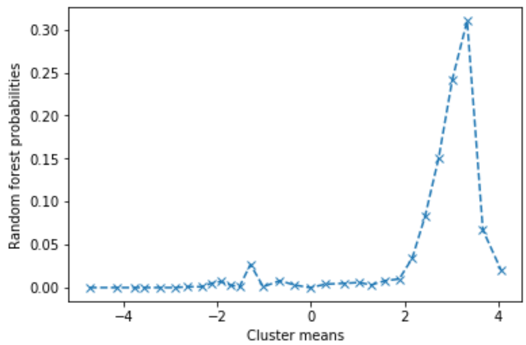

Figure 3.4.3 Probability distribution returned by the RF for the 8th hour of the day 25th of March 2015

At the 95% threshold, the confidence interval of a normal distribution is therefore:

$$]E(X_t) - 1.96\ \sigma;\ E(X_t) + 1.96\ \sigma[$$

Equation 3.4.4

3.4.2 LSTM

Initially the price data were transformed using the same transformation function as presented above, with the exception that the lambda coefficient used here is 100 and not 1 as previously, thus improving the training of the LSTM algorithm. Next, the input data from the time series analysis were reshaped into a matrix. The LSTM algorithm takes as input a vector containing the last 24 values of the series and predicts the next 24. A rolling forecast scenario is used (also called forward model validation). Thus, each time step of the test dataset will be run one at a time.

The parameters used to create the model were designed to find the best compromise between prediction quality and calculation time.

- Batch size: 24
- Nb of hidden LSTM Units: 20
- Nb of epoch (training iterations): 20
- Solver: “Adam”
- Loss function: “MAE” (Mean Absolute Error)

The batch size is necessarily 24 since the algorithm uses 24 data in the past to predict the next 24 data in the future. The number of epochs has been set to 20 since the algorithm improves only slightly over 15/20 iterations as illustrated in Figure 3.4.4. We noticed that the target variable may be mostly Gaussian, but there are outliers (very high or very negative values far from the mean value). The Mean Absolute Error (MAE) loss is an appropriate loss function in this case as it is more robust to outliers. It is calculated as the average of the absolute difference between the predicted and actual values.

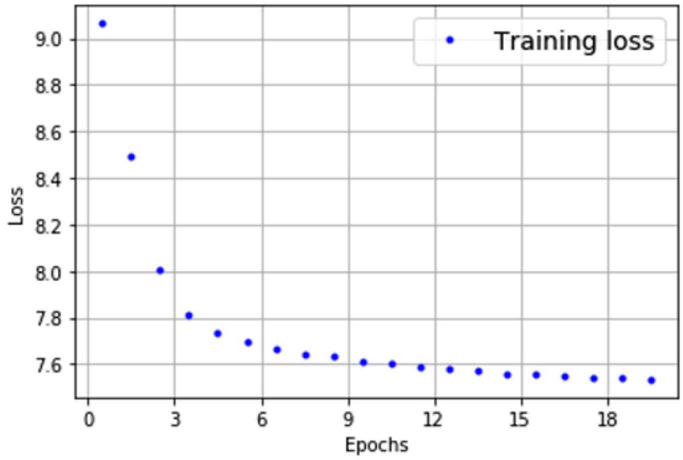

Figure 3.4.4 Convergence of loss function after 20 epochs

4 Results and discussion

In order to compare and analyse the results, we tested our predictive models on three consecutive days from 25th to 27th March 2015. We also calculated errors over the same 30 days for each model, chosen randomly.

4.1 Results of the econometric models

4.1.1 SARIMA results

The SARIMA results are illustrated in Figure 3.1.1. On days 1 and 3, the predictive SARIMA model is really good. It detected very well the pattern of a day. However, on day 2, the model is not performing well. This is due to the fact that day 2 has not a “regular electricity price shape”. The price is much higher during the day than on other days. This might be explained by a lot of different factors: very cold day, few or not any renewable production during this day, a plant broke down… When we take a closer look at the forecasts for each day of the week, we realize that the prediction is particularly bad for Monday and Saturday prices, because there is a significant change in demand between the working days and the weekend and the algorithm bases the prediction on the previous day prices. Thus, it is justified that our SARIMA model could have significant errors for some days. The availability of exogenous data could help this model to better predict prices. The results are indeed presented in the following section.

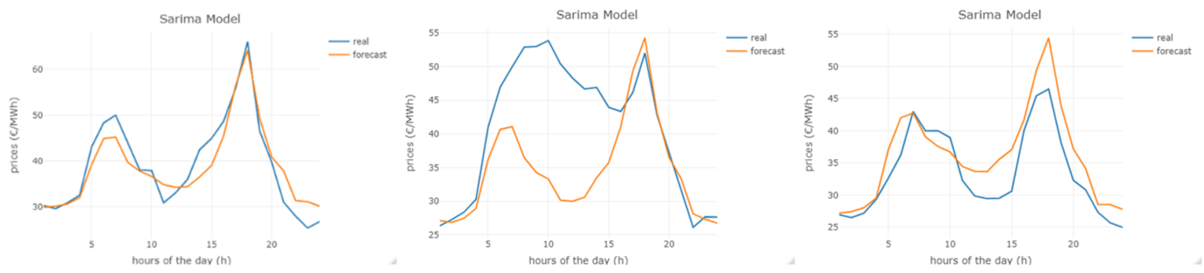

Figure 4.1.2 SARIMA results

4.1.2 SARIMAX results

Exogenous variables (solar & wind production, demand, weekday) were introduced with the linear regression. It is important to remember that these data are realized values and that in reality we would have forecasts, which would add uncertainty to the results. These predictions nevertheless remain fairly reliable over a 24-hour period.

It can be observed with the results in Figure 4.1 that the ordinary linear regression predicts well the trend of the prices. However, the algorithm fails to forecast the price peaks, as on day 1 and 3.

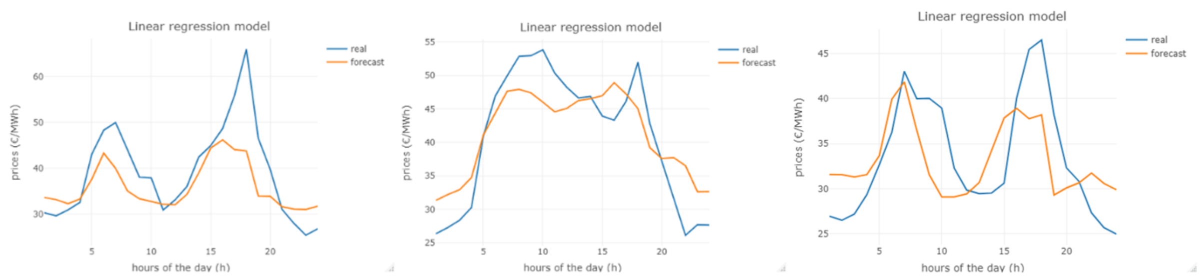

Figure 4.1.3 Linear regression results

As presented above, the idea was then to fit a SARIMA model on the residuals in order to improve the predictions. The results are presented in Figure 4.1.

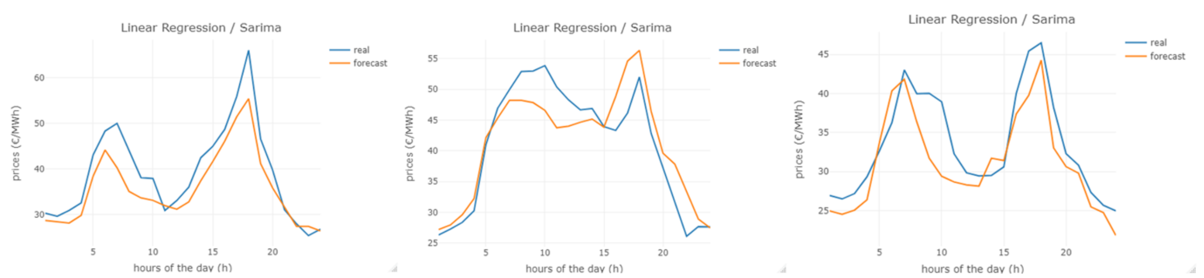

Figure 4.1.4 SARIMAX results

Visually, it is easy to see that applying a SARIMA model on the residuals of the linear regression has greatly improved price forecasting. These results are also much better than those of a classical SARIMA model which is unable to predict atypical days since it does not have additional information brought by exogenous variables.

4.1.3 Holt-Winters results

Some simple commands exist in R to create a prevision thanks to a Holt-Winters method. We decided to use it and to compare the results with our SARIMA model. The drawback of this method is that it only returns a single forecast without any confidence intervals.

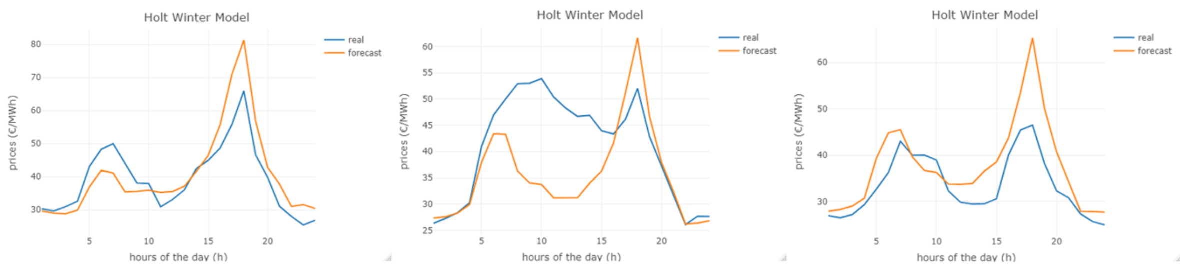

Figure 4.1.5 Holt-Winters results

The results are shown in Figure 4.1. We conclude that this model is quite efficient. Indeed, the results are more than correct considering the fact that this model is extremely simple to put in place. We do not have to make any pre-analysis of our data (except a little transformation in order to get more or less a stationary process). Furthermore, in terms of time processing, it is extremely quick (it takes less than a second) while the SARIMA model previously presented requires approximately one minute to process.

4.1.1 Regime Switching regression model results

The R^2 of the regression is around 0.75 for each regime (0.77 for regime 0 and 0.72 for regime 1). The result of the random forest forecast gives the probability of belonging to each regime, for each time of the day. The forecasts for points that have probabilities close to being in Regime 1 or 2 are often poorer, like for the second chart below. There is also a high variation in prediction when the regime changes, suggesting that we may have forecast errors on intermediate price levels.

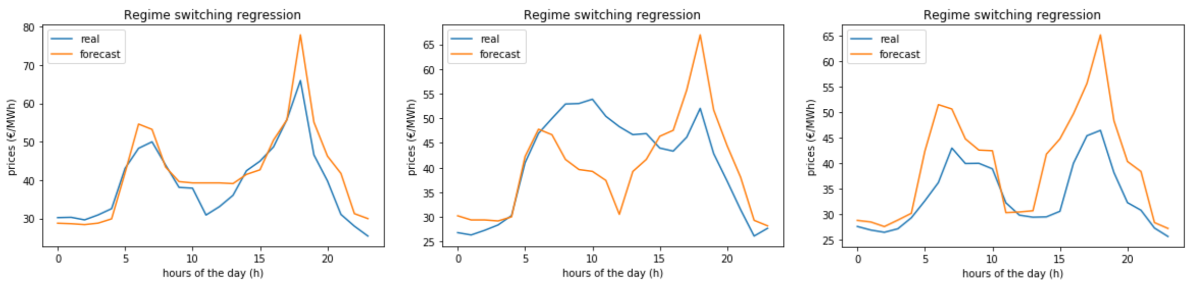

Figure 4.1.1 Regime switching results

4.2 Results of the machine learning models

4.2.1 Random forest results

The results using the RF to forecast the electricity prices of 25th of March 2015 are shown in Figure 4.2.1.

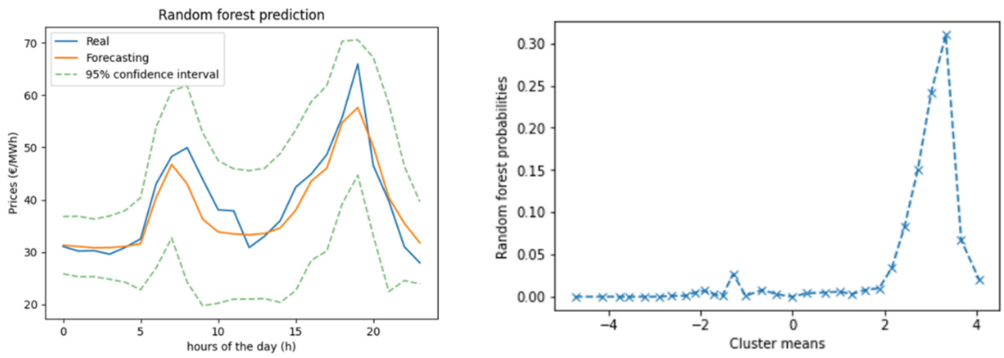

Figure 4.2.1 Left: Prediction and Gaussian confidence intervals (in green) for the 25th of March 2015. Right: Probability distribution returned by the random forest for the 8th hour of the day 25th of March 2015.

The 95% confidence interval in Figure 4.2.1 allows us to see the evolution of the accuracy of the model within the day. For example, we observed that, generally the interval is often thinner for the hours which are well-predicted (especially during the night as the demand does not change from one day to another during these hours). On the contrary, the interval is larger when the prediction is less accurate (particularly around midday). We can surely improve that, by considering the variance of the clusters. This will not have a significant impact on the results because we chose 30 clusters, so the intra-variance is low compared to the inter-variance. The hypothesis of the normal distribution is also wrong for the extreme clusters. Finally, predictions were made from 25th to 27th of March 2015 in Figure 4.2.2. The predictions are well performing for the first and third day and are less accurate for the second day. However, the real prices are almost always included in the 95% confidence interval. We can also notice that the wrong results in the middle of the second day show a very wide confidence interval. Consequently, in practice this large interval may warn that this period of time is very uncertain, and that the probability distribution returned by the random forest is not clear enough as illustrated in Figure 4.2.3.

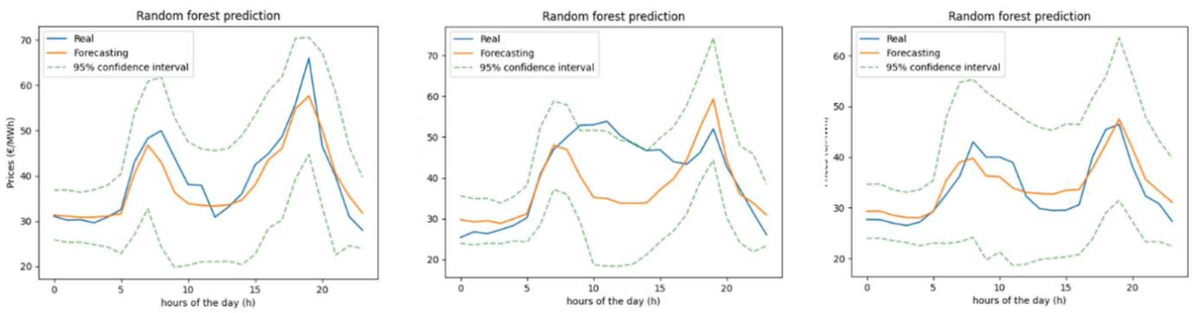

Figure 4.2.2, Real prices versus forecasted prices using a random forest (from 25th to 27th of March 2015)

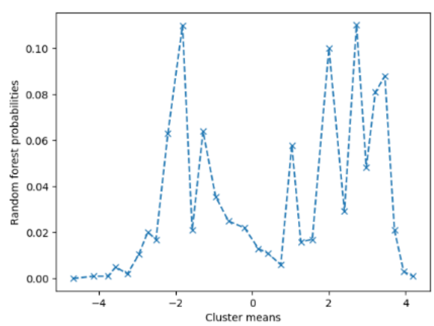

Figure 4.2.3 Probability distribution of the 12th hour of the 26th of March 2015

4.2.2 LSTM results

In order to compare it with our other models, we decided to plot the results on the three days in Figure 4.2.4 previously presented. We used the package keras from the TensorFlow library.

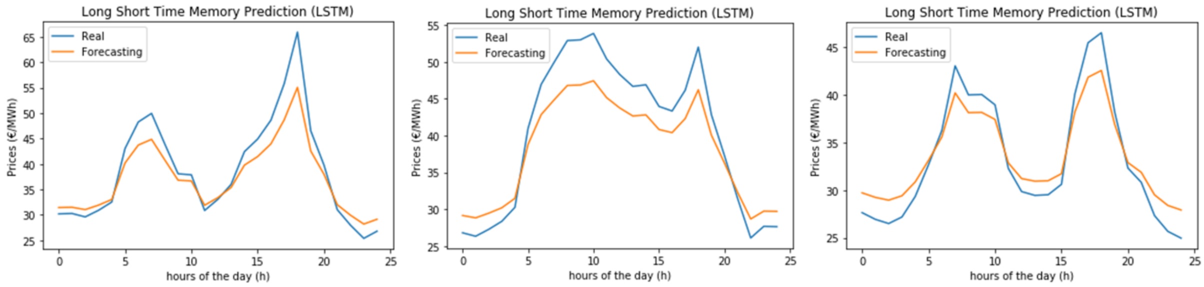

Figure 4.2.4 Real prices versus forecasted prices using LSTM

The results of the LSTM are very good. The advantage of an LSTM model is that it can take into account both a possible autoregression (since it has past values as input variable) and exogenous explanatory variables. It is also possible to be exhaustive in the number of input variables because the model will be able to determine the explanatory variables of interest and "set the rest aside".

4.3 Metrics for comparison

In order to compare the efficiency of our models, we decided to calculate the errors relative to each of these models and to group the results. To do so, we used three error measures: the "Mean Absolute Error - MAE" (in €/MWh) in Equation 4.3.1, the "Root mean squared error – RMSE" in Equation 4.3.2, and finally the "Mean Absolute Percentage Error - MAPE" in Equation 4.3.3, which measures the average relative error of the forecasts (in %).

$$MAE = \sum_{i=1}^{n} |\hat{x}_i - x_i| / n$$

Equation 4.3.1 MAE formula

Where $\hat{x}_i$ is the prediction of x_i made by the model.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{x}_i - x_i)^2}{n}}$$

Equation 4.3.2 RMSE formula

$$MAPE = \frac{1}{n}\sum_{i=1}^{n} |(\hat{x}_i - x_i)/x_i|$$

Equation 4.3.3 MAPE formula

4.4 Results analysis

The values of the different metrics are illustrated in

Table 4.1. The MAPE ranges from around 7% to 20%. The SARIMA and linear regression show similar MAPE of roughly 19%, however it is interesting to note that the SARIMAX model -which results from these two models- exhibits a better MAPE of 13.5%. Indeed, the SARIMAX model correctly forecasts the midday of 26^{th} of March, probably thanks to the solar production provided by the regression. The impact of the solar production is the strongest around noon and this might be reflected in the prices.

	MAE Mean absolute error	**RMSE** Root mean squared error	**MAPE** Mean absolute percentage error
SARIMA	5.27	7.31	19.40%
Linear regression	6.25	7.95	19.60%
SARIMAX	4.05	5.32	13.50%
Holt-Winters	7.36	9.62	13.70%
MS model	4.95	9.33	17.55%
Random Forest	4.54	5.50	15.86%
LSTM	2.63	3.06	6.90%

Table 4.1 Values of the metrics MAE, RMSE and MAPE for each model

The models for which the best forecasting results are obtained (i.e. the models with the lowest MAE, RMSE and MAPE) are in order: LSTM, SARIMAX and Holt-Winters. Time series models such as the SARIMA model or Holt-Winters have very well detected the shape of daily electricity prices. However, they are bad for predicting atypical days or large peaks (both positive and negative). Indeed, as the market is very volatile, prices can vary over a wide range. A SARIMAX model is very interesting because the addition of exogenous variables allows to predict punctual events leading to strong price variations. For example, a surplus of electricity production or an abnormal drop in demand can lead to a strong price decrease. Thus, high price volatility is well captured by models that take into account exogenous variables. Similarly, the RF predicts prices well overall, however the accuracy of the prediction is limited by the number of clusters on which the RF has been trained. Thus, this model seems less adapted than SARIMAX for a very volatile market.

5 Conclusion

In this research work we have implemented different models for forecasting short-term electricity prices on the German market according to two main approaches: an econometric/time series analysis approach (SARIMA, SARIMAX, Holt-Winters and Markov Switching Model), and a machine learning approach (Gaussian Mixture and LSTM). The SARIMA and Holt-Winters models do not consider exogenous variables and are unable to predict atypical days. These models have nevertheless the merit of being very simple to set up while producing correct results since they predict relatively well the classical days. The SARIMAX model gives better results than the previous models since it can capture additional information (volume, solar, wind) that allow it to predict atypical days. The simplified version of the Markov Switching model gives slightly less satisfactory results but is still correct. The results from the machine learning methods are very satisfactory, especially those of the LSTM. The RF does not perform as well as the SARIMAX however it returns a confidence interval that widens when prices are harder to predict around midday. The main advantage of machine learning methods is that one can be exhaustive on the input variables and one could still improve the performance of these models by entering other exogenous variables such as the prices of different commodities (oil, gas, coal, uranium, ...) as well as other information such as the price of CO2 and temperature forecasts. Application of models such as those presented here is essential for many companies involved in the energy sector for their activities of buying and selling electricity on spot markets, hedging and risk management, scheduling the production of a power plant, etc. Several limits of our models could be dealt in a further research work. Analyzing the confidence intervals of our econometric methods would help companies to make decisions with a certain level of confidence. In order to complete this first work, for example it would be interesting to create GARCH models that would take into account the fact that the transformed price variable is not rigorously homoscedastic. In addition, it would be also interesting to have access to many exogenous variables that could enrich these models and improve the prediction results: e.g. fuel costs (oil, gas, uranium, coal), CO2 costs, power plant availability, filling level of water dams, temperature forecasts, risks of congestion on the interconnection network…

Bibliography

[1] K. Maciejowska, W. Nitka and T. Weron, "Day-ahead vs. Intraday—Forecasting the price spread to maximize economic benefits.," *Energies ,* vol. 12.4, p. 631, 2019.

[2] F. Paraschiv, D. Erni and R. Pietsch, "The impact of renewable energies on EEX day-ahead electricity prices.," *Energy Policy,* vol. 73, pp. 196-210, 2014.

[3] P. Deane et al., "Quantifying the" merit-order" effect in European electricity markets," 2015.

[4] T. Pham and K. Lemoine, "Impacts of subsidized renewable electricity generation on spot market prices in Germany: evidence from a garch model with panel data.," 2020.

[5] "Smard," [Online]. Available: https://www.smard.de/en/datenschutz.

[6] O. Renn and J. P. Marshall, "Coal, nuclear and renewable energy policies in Germany: From the 1950s to the "Energiewende," *Energy Policy,* vol. 99, pp. 224-232, 2016.

[7] "Germany 2020," IEA, Paris, 2020.

[8] Stylianos I. Vagropoulos, G. I. Chouliaras, E. G. Kardakos, C. K. Simoglou and A. G. Bakirtzis, "Comparison of SARIMAX, SARIMA, modified SARIMA and ANN-based models for short-term PV generation forecasting," *IEEE International Energy Conference,* vol. 45, no. 21, pp. 1-6, 2016.

[9] V. G. Tran, V. Debusschere and S. Bac, "One week hourly electricity load forecasting using Neuro-Fuzzy and Seasonal ARIMA models," *IEEE International Energy Conference,* vol. 8, pp. 97-102, 2013.

[10] M. Olsson and . L. Soder, "Modeling Real-Time Balancing Power Market Prices Using Combined SARIMA and Markov Processes," *IEEE Transactions on Power Systems,* vol. 23, no. 2, 2008.

[11] N. Elamin and M. Fukushige, "Modeling and Forecasting Hourly Electricity Demand by SARIMAX with Interactions," *Energy,* vol. 165, pp. 257-258, 2018.

[12] P. Ďurka and S. Pastoreková, "ARIMA vs. ARIMAX – which approach is better to analyze," *International Conference Mathematical Methods in Economics,* 2012.

[13] Brown and R. Goodell, "Smoothing, forecasting and prediction of discrete time series.," *Courier Corporation,* 2004.

[14] R. E. Q. Stephen M. Goldfeld, "A Markov model for switching regressions," *Journal of Econometrics,* vol. 1, no. 1, pp. 3-15, 1973.

[15] J. D. Hamilton, "Analysis of time series subject to changes in regime," *Journal of Econometrics,* vol. 45, no. 1–2, pp. 39-70, 1990.

[16] F. S. N. M. Ø. N. Niels Haldrup, "A vector autoregressive model for electricity prices subject to long memory and regime switching," *Energy Economics,* vol. 32, no. 5, pp. 1044-1058, 2010.

[17] M. Ø. N. Niels Haldrup, "A regime switching long memory model for electricity prices," *Journal of Econometrics,* vol. 135, no. 1–2, pp. 349-376, 2006.

[18] A. Cifter, "Forecasting electricity price volatility with the Markov-switching GARCH model: Evidence from the Nordic electric power market," *Electric Power Systems Research,* vol. 102, pp. 61-67, 2013.

[19] G. Zachmann, "A stochastic fuel switching model for electricity prices," *Energy Economics,* vol. 35, pp. 5-13, 2013.

[20] Y. N. X. C. Timothy D. Mount, "Predicting price spikes in electricity markets using a regime-switching model with time-varying parameters," *Energy Economics,* vol. 28, no. 1, pp. 62-80, 2006.

[21] R. Huisman, "The influence of temperature on spike probability in day-ahead power prices," *Energy Economics,* vol. 30, no. 5, pp. 2697-2704, 2008.

[22] V. A.E.D., “Modelling the Impact of Wind Power Production on Electricity Prices by Regime-Switching Lévy Semistationary Processes.,” pp. In: Benth F., Di Nunno G. (eds) Stochastics of Environmental and Financial Economics. Springer Proceedings in Mathematics & Statistics, vol 138. Springer, Cham. https://doi.org/10.1007/978-3-319-23425-0_13, 2016.

[23] C. Martin de Lagarde and F. Lantz, “How renewable production depresses electricity prices: Evidence from the German market.,” *Energy Policy,* vol. 117, pp. 263-277, 2018.

[24] Y. Huang et al., “A Gaussian mixture model based classification scheme for myoelectric control of powered upper limb prostheses.,” *IEEE Transactions on Biomedical Engineering,* vol. 52.11, pp. 1801-1811, 2005.

[25] D. J. Swider and C. Weber, “Extended ARMA models for estimating price developments on day-ahead electricity markets.,” *Electric Power Systems Research,* Vols. 77.5-6, pp. 583-593, 2007.

[26] F. Dellaert, “The expectation maximization algorithm,” *Georgia Institute of Technology,* 2002.

[27] J. Mei et al., “A random forest method for real-time price forecasting in New York electricity market,” *General Meeting| Conference & Exposition. IEEE,* 2014.

[28] A. Lahouar and J. B. H. Slama, “Day-ahead load forecast using random forest and expert input selection.,” *Energy Conversion and Management,* vol. 103, pp. 1040-1051, 2015.

[29] A. Sadeghi-Mobarakeh et al., “Data mining based on random forest model to predict the California ISO day-ahead market prices.,” *IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT),* 2017.

[30] P. O. Gislason, J. A. Benediktsson and J. R. Sveinsson, “Random forests for land cover classification.,” *Pattern Recognition Letters,* vol. 27.4, pp. 294-300, 2006.

[31] F. Rosenblatt, “The perceptron: a probabilistic model for information storage and organization in the brain.,” *Psychological review,* vol. 65.6, p. 386, 1958.

[32] K. Gurney, “An introduction to neural networks,” *CRC press,* 1997.

[33] Bengio, Yoshua, P. Simard and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult.,” *IEEE transactions on neural networks,* vol. 5.2, pp. 157-166, 1994.

[34] Pascanu, Razvan, T. Mikolov and Y. Bengio, “On the difficulty of training recurrent neural networks.,” *International conference on machine learning,* 2013.

[35] Hochreiter, Sepp and J. Schmidhuber, “Long short-term memory,” *Neural computation,* vol. 9.8, pp. 1735-1780, 1997.

[36] Sewalt, Michael and C. D. Jong, “Negative prices in electricity markets.,” *Commodities Now,* vol. 7, pp. 74-77, 2003.

[37] Johnson and N. L., “Systems of frequency curves generated by methods of translation,” *Biometrika,* vol. 36.1/2, pp. 149-176, 1949.

[38] Schneider and Stefan, “J. Energy Mark.,” *Power spot price models with negative prices,* p. 77–102, 2012.

[39] H. e. a. Cook, "CUDA-level Performance with Python-level Productivity for Gaussian Mixture Model Applications.", 2011.

[40] Pedregosa et al., “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research,* vol. 12, pp. 2825-2830, 2011.

Retrouvez toute la collection
https://www.ifpenergiesnouvelles.fr/article/les-cahiers-leconomie

228 - 232 avenue Napoléon Bonaparte
92852 Rueil-Malmaison
www.ifpschool.com

1-4 avenue de Bois-Préau
92852 Rueil-Malmaison
www.ifpenergiesnouvelles.fr