DrahiX Power Consumption Forecast

Oscar Courbit, Ziad El Assal

March 9, 2022

Abstract

Power consumption forecasting plays a key role in the energy sector for adapting the power supply to the demand. The rise of renewable energies, whose main drawback is intermittence, makes forecasting even more important. At a smaller scale, power consumption forecasting is crucial in areas that have their own source of energy such as solar panels. In this project, we forecast the power consumption of the DrahiX, Ecole Polytechnique's entrepreneurship building, based on past measurements and exogenous meteorological data. We use linear models such as ARIMA, and Prophet, that give good results for prediction horizons going from hours to 5 days.[1]

Contents

[1] Code available on https://github.com/ziadelassal/map512

1 Problem formulation

The objective of this project is to predict the energy consumption of a building, having as initial data this consumption as well as external weather data over a period of more than two years. There are some difficulties regarding the exploitation of the energy data: firstly, the range of data we have has been marked by repeated lockdowns that do not represent the usual consumption of the building under normal conditions. The other difficulty with this project is that it is very large and there can be many issues:

- Do we want to predict the minute-by-minute consumption, as shown in our data, or do we want to predict the trajectory of the hour-by-hour evolution of that consumption?
- Is the objective to predict in the very short term (by predicting for example the consumption during the next hour by looking at the rest of the day) or to see in the very long term (by predicting the consumption over a whole month by looking at the rest of the year)?

Our intuition was to think about a medium-term consumption prediction, trying to predict between one and five days of consumption based on previous data. We also noticed that using too much upstream data to train would have a significant cost for running our models and would not gain much accuracy. Therefore, we set up models training on only a few weeks and predicting only a few days of data.

2 Previous work and models used

Forecasting in time series is an important area of machine learning. Time series analysis is present in many business fields such as energy, finance, biology, public policy or business. It has motivated the writing of many books and surveys. We heavily used the work of R. Nau (2016)[1] for ARIMA-related models, as well as R. Hyndman and G. Athanasopoulos (2021) [2] book. We also relied on research papers, namely Facebook's paper (2017) for the Prophet model.[3]

2.1 AutoReg

Contrary to multiple regression models, autoregressive models forecast the variable using a linear combination of its past values rather than of predictors. An autoregressive model of order p, AR(p), is written as :

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p} + \epsilon_t, \tag{1}$$

where ϵ_t is the error. Autoregressive models are very flexible and can handle many different time series types.[2]

2.2 ARIMA

Stationarity and Differencing A **stationary** time series is a time series with time-independent properties. For instance, trends and seasonalities make a time series non stationary. **Differencing** is a way to stabilize the mean and the variance of a time series to make it stationary. It consists in computing the difference between consecutive values of the variable, at different degrees d. Let y'_t be the differences time series y_t at time t. If $d = 0$, $y'_t = y_t$. If $d = 1$, $y'_t = y_t - y_{t-1}$. If $d = 2$, $y'_t = (y_t - y_{t-1}) - (y_{t-1} - y_{t-2})$.[1]

ARIMA ARIMA (AutoRegressive Integrated Moving Average) is a combination of an autoregressive and a moving average (MA) model. MA models use past errors ϵ_i rather than past values to predict the value of the variable. A moving average model MA(q) of order q can be written as:

$$y'_t = c + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + ... + \theta_q \epsilon_{t-q}. \tag{2}$$

Thus, an **ARIMA**(p, d, q) model is written as :

$$y'_t = c + \phi_1 y'_{t-1} + ... + \phi_p y'_{t-p} + \theta_1 \epsilon_{t-1} + ... + \theta_q \epsilon_{t-q} + \epsilon_t, \tag{3}$$

where y'_t is the differenced series, p the autoregression order, d the degree of differencing needed for stationarity, and q the moving average order.[2]

In order to choose the optimal values for p and q, we use the **ACF** (auto-correlation function) and **PACF** (partial auto-correlation function) plots. The ACF computes the correlation of a series with its lagged values, i.e. between y_t and y_{t-k} for different values of k, while the PACF plots the correlations between the residuals that remain after removing the effects of correlation due to earlier values, and that are already explained by the ACF. The partial auto-correlation at lag k is equal to the k^{th} estimated coefficient in an AR(k) model. It reflects the effect of y_{t-k} after removing the effects of the $k-1$ first lags. Thus, we use the PACF to determine how many AR terms are needed: if the PACF cuts off at lag k, we should try an AR(k) model. Note that if both p and q are positive, the plots do not help to find optimal values of p and q.[1]

2.3 SARIMAX

We also use a variation of ARIMA, namely SARIMAX (Seasonal Auto-regressive Integrated Moving Average models with exogenous factor) in order to take the seasonality effects into account.[4] It is usually written as SARIMAX$(p,d,q)\times(P,D,Q)_s$, where P is the number of seasonal auto-regressive terms, D the number of seasonal differences, Q the number of seasonal moving average terms, and s the seasonal period. The seasonal terms are similar to the non-seasonal terms except that they involve backshifts to the seasonal period.[2] Besides, the integration of the exogenous factors will be useful as some variables (e.g. the temperature) are highly correlated with the power consumption.

2.4 Prophet

Prophet is a forecasting tool developed by Facebook's (now Meta) Core Data Science team. It is based on an additive model that takes into account yearly, weekly and daily seasonalities, along with holiday effects. The initial motivation of Prophet came from the fact that "*the demand for high quality forecasts often far outstrips the pace at which they can be produced*"[3], hence the production of a forecasting tool *at scale*.

Prophet uses a decomposable time series model (Harvey & Peters, 1990) with tree components, trend, seasonality and holidays:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t, \tag{4}$$

where $g(t)$ is the trend function (it models non periodic changes), $s(t)$ represents periodic changes and $h(t)$ represents the effects of holidays. ϵ_t represents any changes that are not accommodated by the model. Two of the advantages of Prophet that are particularly useful in our case are that, first, it fits very fast, allowing to explore the model and to adjust parameters, and second, is has interpretable parameters that can be changed easily.

Trend The trend, in its simplest form, is defined as

$$g(t) = \frac{C}{1+\exp\left(-k(t-m)\right)}, \tag{5}$$

where C is the carrying capacity (for instance, the maximum power consumption), k the growth rate and m an offset parameter.[3] The trend model can be complexified by incorporating trend changes, given that C and k are generally not constant. Prophet automatically detects changepoints and models a linear trend between them.

Seasonality Time series often show a daily, weekly, monthly or yearly seasonality pattern. Prophet rely on Fourier series to model periodic changes. If P is the period expected for the time series, seasonal effects can be approximated by

$$s(t) = \sum_{n=1}^{N} a_n \cos\left(\frac{2\pi nt}{P}\right) + b_n \sin\left(\frac{2\pi nt}{P}\right), \tag{6}$$

that we can rewrite as

$$s(t) = X(t)\boldsymbol{\beta}, \tag{7}$$

where $X(t) = \left[\cos\left(\frac{2\pi t}{P}\right), ..., \sin\left(\frac{2\pi n t}{P}\right)\right]$ and $\boldsymbol{\beta} = [a_1, b_1, ..., a_n, b_n]^\top$ that has to be estimated. In the generative model, i.e. for prediction, Prophet takes $\boldsymbol{\beta} \sim \mathcal{N}(0, \sigma^2)$. By varying N, we can define an arbitrary smooth seasonal effect.

Holidays and Events Prophet has an integrated database for holidays. For each holiday i, let D_i be the list of dates for this holiday. Then we can write the holiday component, as in 7, as

$$h(t) = Z(t)\boldsymbol{\kappa} \tag{8}$$

where $Z(t) = [\mathbf{1}_{t \in D_1}, ..., \mathbf{1}_{t \in D_L}]$. For the generative model, $\boldsymbol{\kappa} \sim \mathcal{N}(0, \nu^2)$.

As we will see it below, an important advantage of the decomposable model is that we can look at each component separately, gaining in interpretability.[3]

3 Exploratory data analysis

3.1 The dataset

We are provided with a dataset containing the folllowing variables:

- Date & Time

Electric power measurements in Zone 2 of Drahi-X building:

- Usage [kW] : Total consumption of Zone 2 (= "Total Zone 2 [kW])
- Generation [kW] : Empty data (there is no generation data in Zone 2)
- SUM [kW] : Sum of the majority of consumption in Zone 2 (="Heating/Cooling Total" + Lights + Plugs + WaterHeater + Ventilation)
- Plugs [kW] : Electric consumption of electric plugs in Zone 2
- Heating / Cooling Total [kW] : Total consumption of Heating and cooling. It includes Heatpump consumption + the electric heaters ("Heaters corridor" and "Heaters Toilets")
- Ventilation [kW] : Electric consumption of air ventilation
- Heaters corridor [kW] : Electric consumption of heaters that are in the corridor (each heater has 500W of power)
- Lights [kW] : Electric consumption of lights
- Water heater [kW] : Electric consumption of water heater (nominal power : 1200 W)
- Heaters Toilets [kW] : Electric consumption from heaters that are in the toilets
- Total Zone 2 [kW] : Total electric consumption of Zone 2 (= "Usage [kW]")

Extra time variables:

- Time in sec
- Weekday
- Time of day

SIRTA's meteorological measurements:

- Direct_Solar_Flux : Direct normal irradiance (W/m^2)
- Diffuse_Solar_Flux : Diffuse solar irradiance on a horizontal surface (W/m^2)

- Global_Solar_Flux : Global solar irradiance on a horizontal surface (W/m^2)
- Downwelling_IR_Flux : Long-wave (infrared) downwelling radiation (W/m^2)
- SZA : Solar Zenith Angle (deg)
- SAA : Solar Azimuth Angle (deg, South=0, East=-90, West=90)
- ws : wind speed at 10m above the ground (m/s)
- wd : wind direction at 10m above the ground (deg, North=0, East=90)
- AirTemp : Air temperature (^oC)
- rh : air relative humidity (%)
- pres : air pressure (hPa)
- rain : precipitation (mm)

3.2 Data visualization

First we observe the Power consumption evolution during a complete year along with its distribution. We smooth the data by using a rolling window and applying to each value the mean of the values surrounding it. The size of the rolling window can vary. With a rolling window of size 1440 minutes (one day), we observe the weekly seasonalities (Fig. 1).

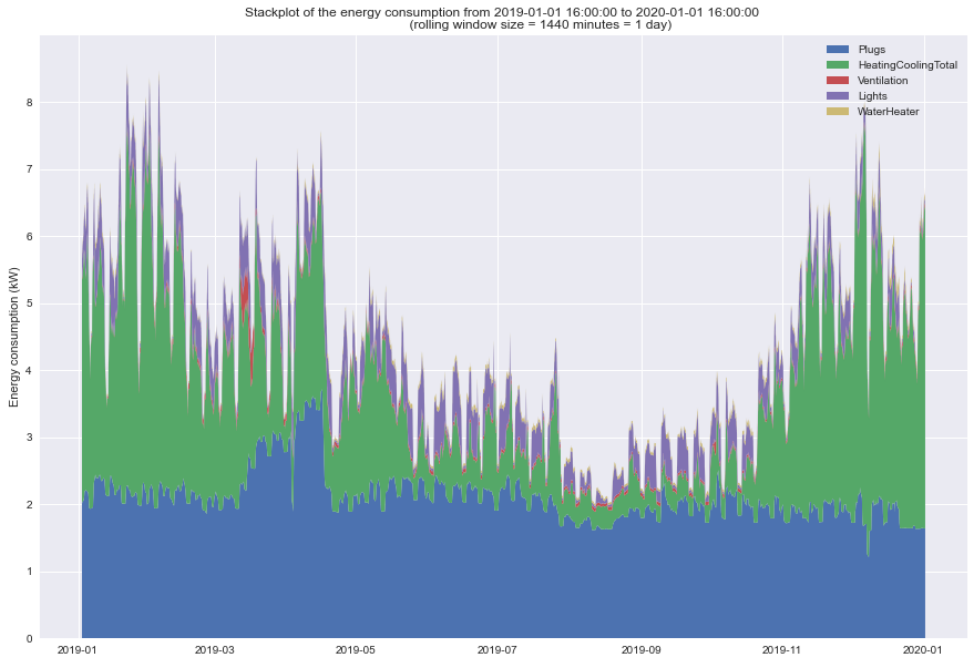

Figure 1: Stackplot of the power consumption during year 2019 (one-day rolling window).

The total consumption has a few peaks during the year (February, April, December) and has a global minimum in August, probably related to summer holidays. We then use a one week rolling window to better visualize the general trend and the distribution (Fig. 2). Plugs and heating/cooling represent the greatest part of the power consumption. While the consumption related to plugs remain approximately constant over the year (except for a peak between March and April 2019), the heating/cooling goes down between June and October, with higher temperatures (since heating consumes more energy than cooling). Aside, lights,

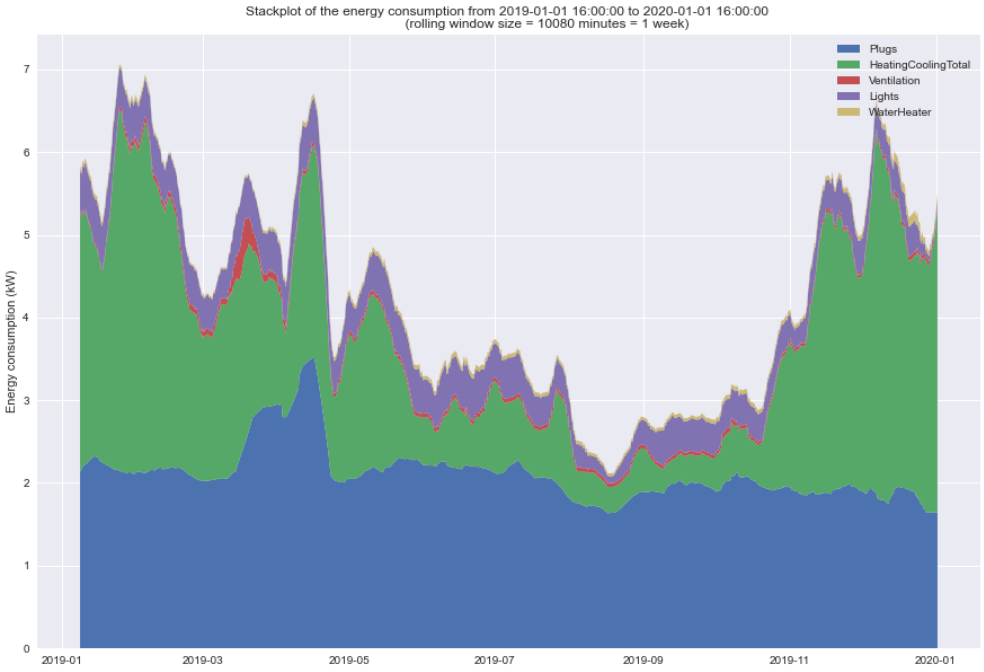

Figure 2: Stackplot of the power consumption during year 2019 (one-week rolling window).

ventilation and water heaters consumption remain roughly constant over the year, and represent a small part of the total consumption.

We then observe the consumption in a period of 2 months to search for other trends (Fig. 3), here during October and November. We observe the daily seasonalities with power consumption going down during the weekends. It is interesting to note that there is a constant flow of energy from plugs (about 1.9 kW) which adds up to the seasonal flow. We can take advantage of this information later.

4 Forecasting

4.1 Feature selection and engineering

In our problem, the feature engineering and selection process is quite straightforward. We are already provided with a useful Weekday variable that helps the model take into account the weekly seasonalities. Given the particular pattern on weekends, we have added a binary Weekend variable that will be helpful to non-seasonal models like Autoreg. We then choose the relevant exogenous variables, first by examining the correlation matrix then by testing their effects directly with the models.

We display to correlation matrices: one with a one-hour sampling (Fig. 4), which is the sampling we usually use for prediction, and another with an every-day sampling (Fig. 5).

They both display an obvious positive correlation between the power consumption-related variables, some of which are redundant. We focus on the variable Usage. It has a high (positive or negative) correlation with the variables AirTemp, Downwelling_IR_Flux and rh. Without surprise, AirTemp is itself highly correlated with the humidity rh (negatively) and with Downwelling_IR_Flux (positively).

We choose to keep AirTemp and Downwelling_IR_Flux, and Weekend or Weekday according to the model.

Given the high noise level of the data, we applied a rolling mean to the time series in order to smoothen it. We also sampled it at a lower frequency, both because we don't need an every-minute sampling for our forecast scale (a day or more), and for computational speed. Most of the time, we use a rolling window and a sampling step of 60 minutes.

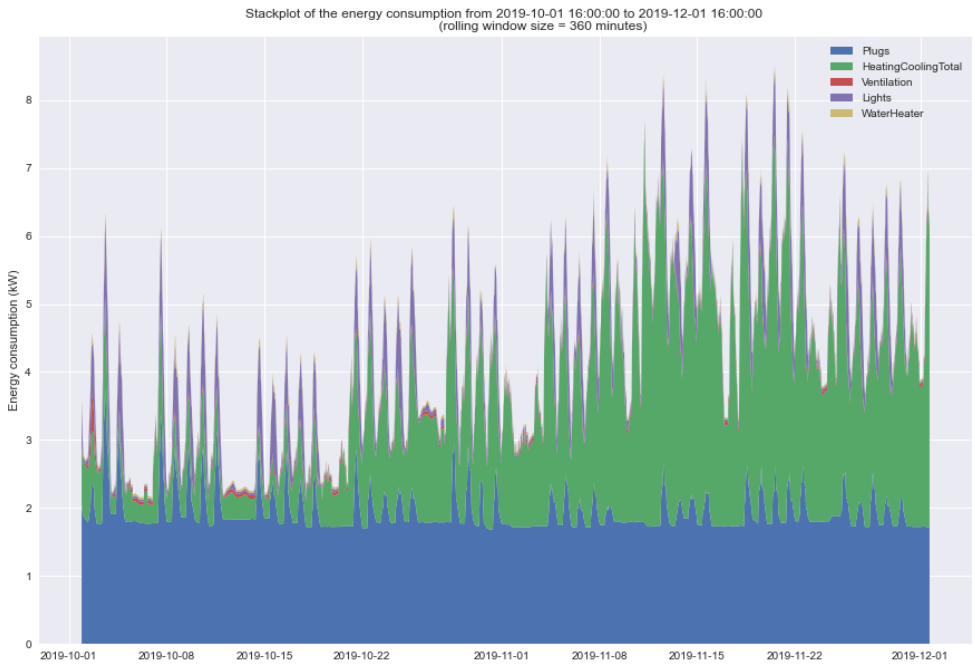

Figure 3: Stackplot of the power consumption during October and November 2019 (one-week rolling window).

Choice of dates In the first instance, we choose the following dates to test the models:

- Fitting: **2019-09-09 00:00:00** to **2019-09-26 00:00:00**
- Forecasting: **2019-09-26 00:00:00** to **2019-10-01 00:00:00**

These periods have the advantage of being complete (no holes in the data), and they include weekends, that have a specific regime and thus are more difficult to predict (Fig. 6). These dates are representative of the entire time series.

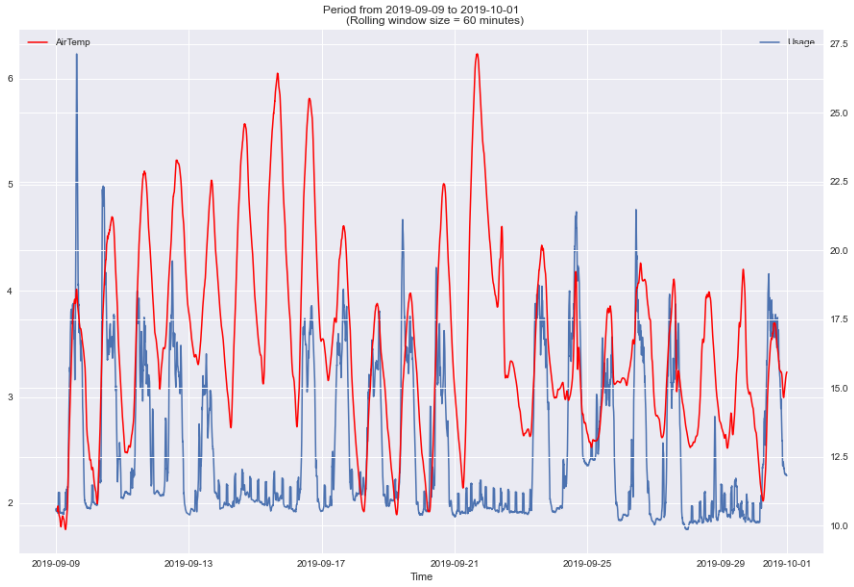

Figure 6: Plot of the variables Usage and AirTemp during the chosen period

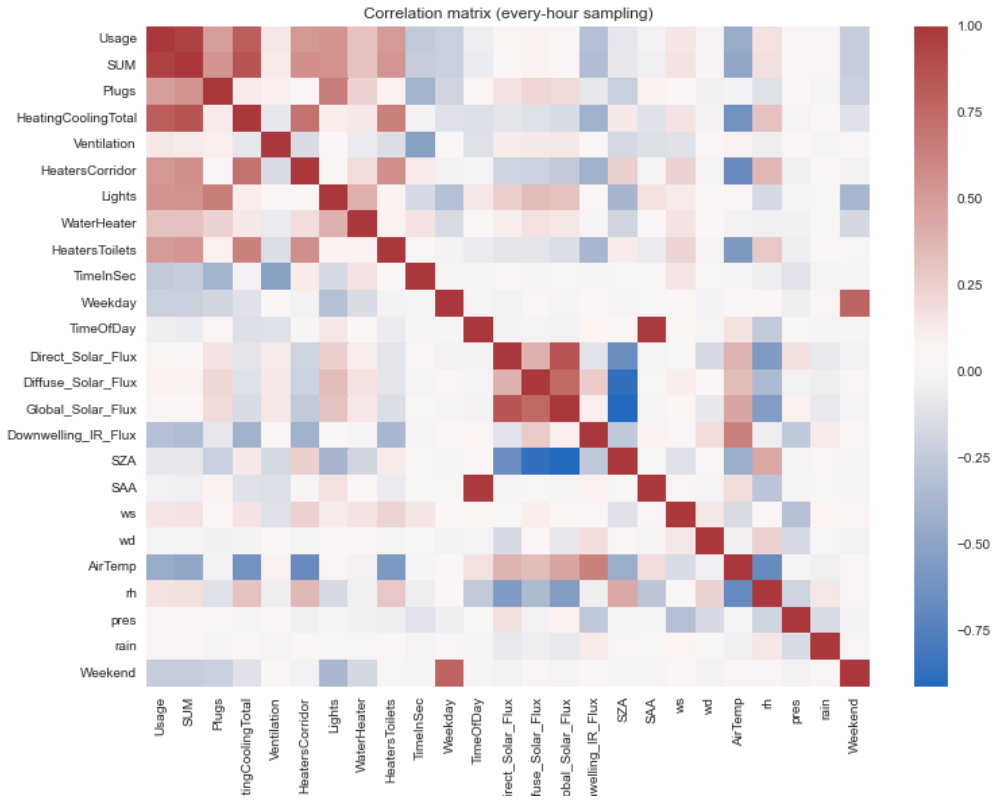

Figure 4: Correlation matrix between the variables with an every-hour sampling.

4.2 Forecasting

4.2.1 AutoReg

As AutoReg prediction is based on the linear combination of the previous values, the main parameter to adjust is the order p of the AR(p) model, which refers to the number of previous values to be integrated in the linear combination. Increasing the order first enhances the predictions, but adding lags also increases the noise and the risk of overfitting. Choosing the number of lags is finding the right equilibrium between smoothness and accuracy.

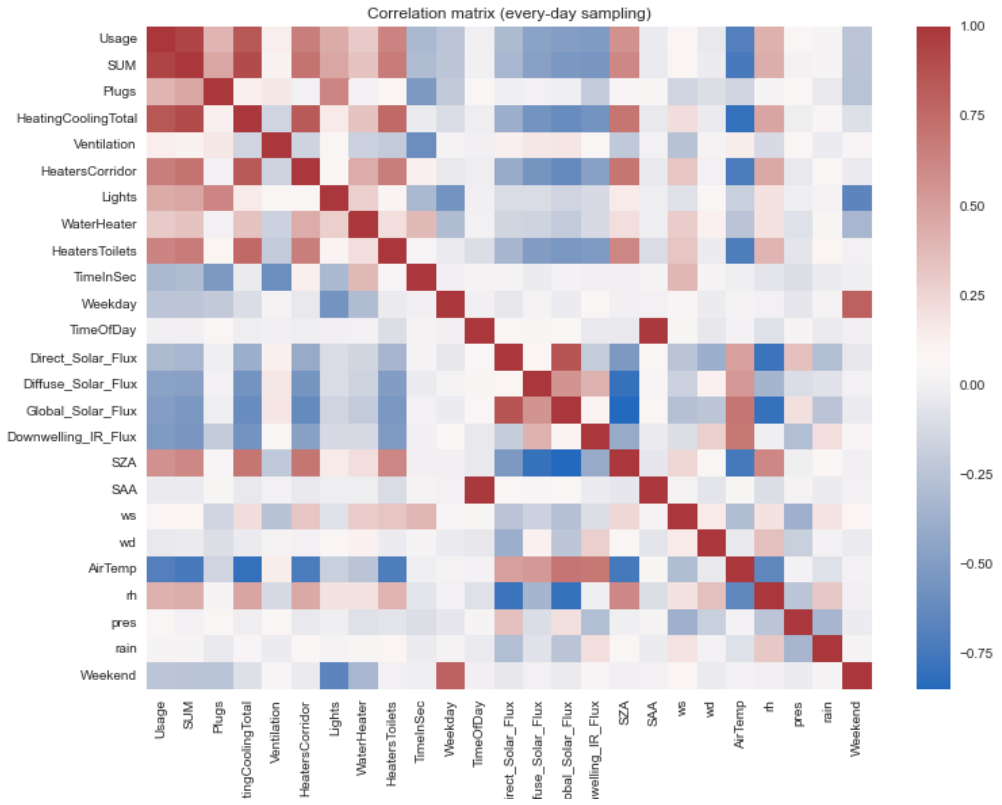

Figure 5: Correlation matrix between the variables with an every-day sampling. The correlations are globally more important than for the every-hour sampling, which shows that exogenous variables has a stronger impact on the heating usage at low-frequency. For instance, the variation of the temperature has more impact on energy usage on a yearly scale than on a daily scale, since the difference of heating between summer and winter is higher than at different moments of a single day.

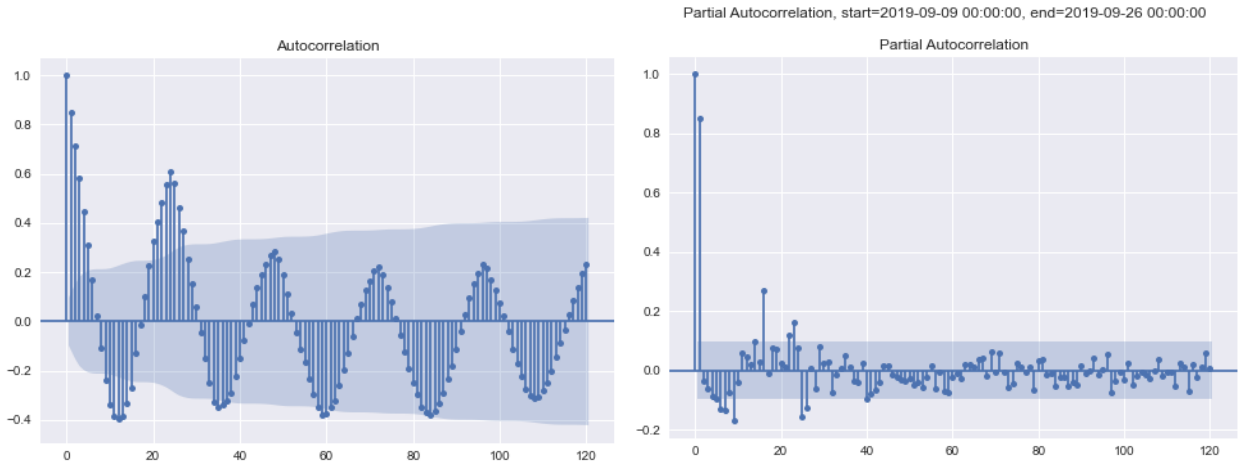

Figure 7: ACF and PACF plots with a 60-minutes sampling, $\alpha = 0.05$. On the ACF, the daily seasonality is visible. The PACF displays 3 correlation peaks at +9h, +16h and +24h, meaning that they are not explained by the AR model. There is no peak after 24 hours.

The order p can be determined by analyzing the ACF and PACF plots. As mentioned in 2.2, the partial

auto-correlation at lag k is equal to the k^{th} estimated coefficient in an AR(k) model. On Fig. 7, a cutoff appears after 24 hours, which reflects the daily seasonality. Therefore, we first use a simple AR(24) model (Fig. 8). The forecast displays a daily seasonality but it fails to forecast the right amplitude and performs poorly on weekends, given their special regime.

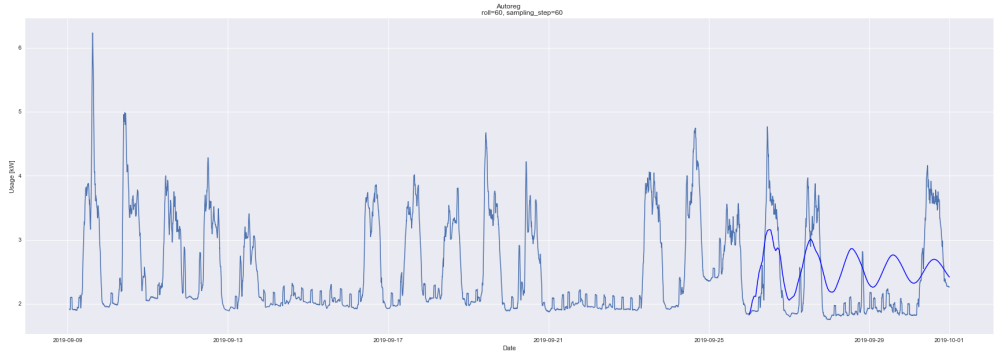

Figure 8: Forecasting with an AR(24) model. MAE: 0.59, MSE: 0.52

To tackle the problem, we first extend the order to 96 (4 days) (Fig. 9), then we add the exogene binary variable Weekend, which value is 1 if the day is a Saturday or Sunday, and 0 if not (Fig. 10)

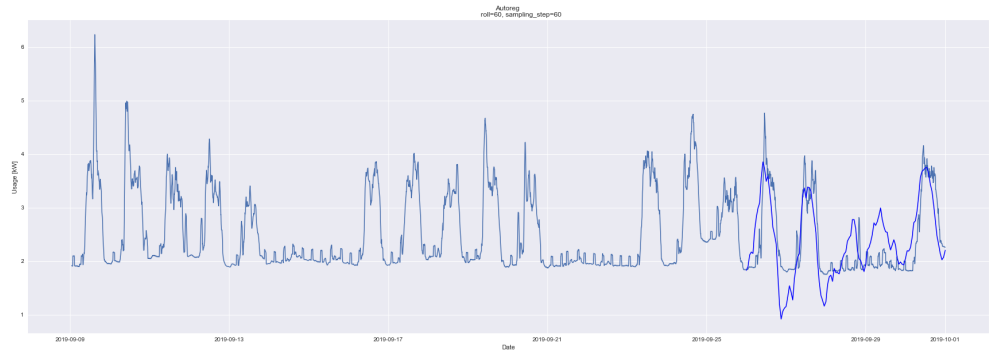

Figure 9: Forecasting with an AR(96) model. MAE: 0.45, MSE: 0.35

Then we have chosen the most relevant exogenous values for this model (Fig. 10): the most determining for AutoReg is the column "Weekend" indicating with a boolean if the considered date is in a weekend or not. We also took into account the air temperature ("AirTemp" column) which, according to the correlation matrix (Fig. 4), has some influence on the energy consumption. Adding other exogenous values does not really improve the results and the calculations become more expensive.

It should be noted that the use of exogenous values becomes complex for AutoReg if we include the dates of the hour changes (twice every year). This is due to the specificity of this algorithm; it will not be a problem for the other models.

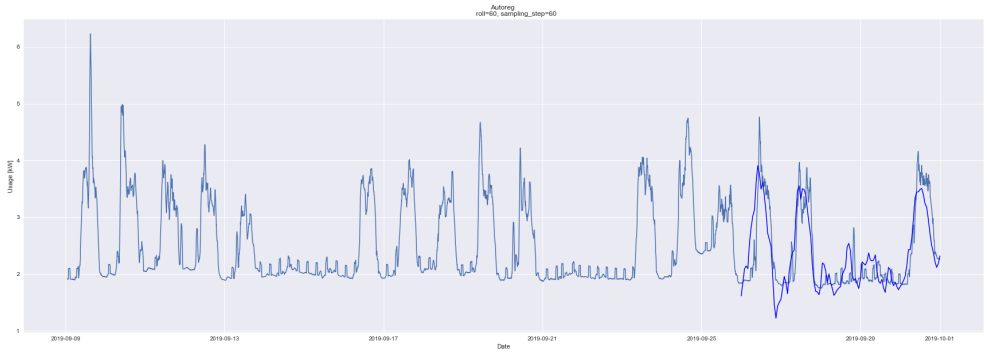

Figure 10: Forecasting with an AR(96) model with exogenous features. MAE: 0.34 , MSE: 0.27

4.2.2 ARIMA

Compared to AR, ARIMA has a differenciating and a moving average components. The use of the moving average did not provide better results. The best results we have obtained are for the parameters $(48, 1, 0)$, i.e. with one order of differencing and no moving average. ARIMA will be completed by the use of SARIMAX, which will better consider the weekends (weekly seasonality) and exogenous factors (Fig. 11).

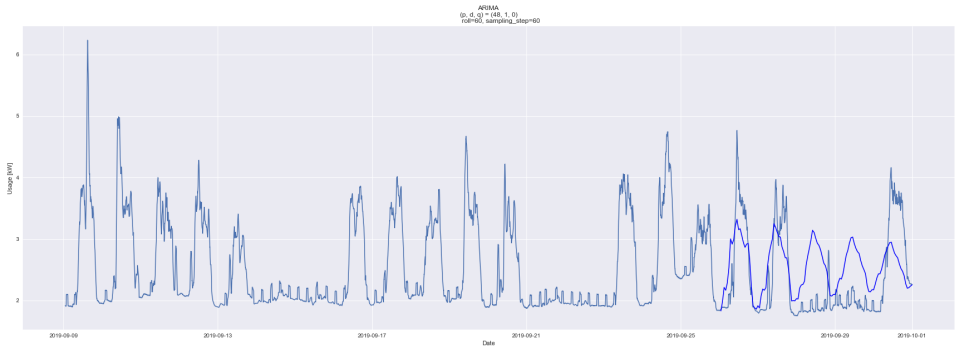

Figure 11: Forecasting with an ARIMA$(48, 1, 0)$ model. MAE: 0.57, MSE: 0.53

4.2.3 SARIMAX

SARIMAX uses two more pieces of information than ARIMAX:

- the seasonality of the model (with the seasonal orders),
- the exogenous values.

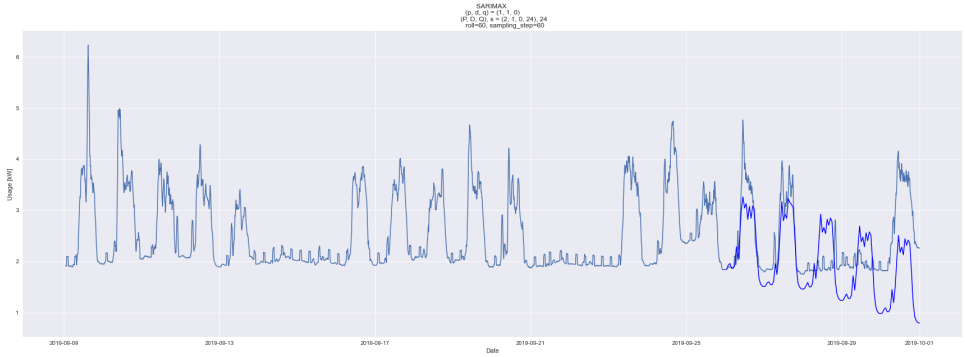

Figure 12: Forecasting with a SARIMAX$(1,1,0)(2,1,0)_{24}$ model. MAE: 0.65, MSE: 0.71

Without these two pieces of information, SARIMAX gives the same results as an ARIMA model (Fig. 11. This is why the seasonal parameter s is very useful: we first tried to use a daily seasonality ($s = 24$). The model performs poorly, has an unjustified decreasing trend and does not consider the weekends (Fig. 12). We then used a weekly seasonality ($s = 7 \times 24 = 168$). The model performed way better, considering very well the presence of the weekends in the testing data as well as a vaild trend (Fig. 13).

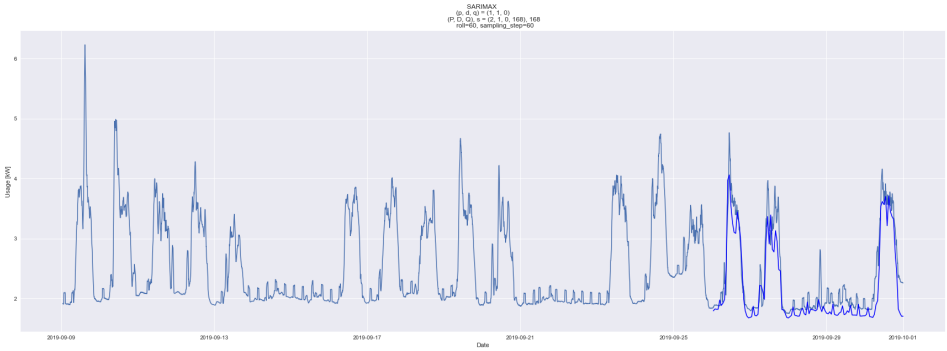

Figure 13: Forecasting with a SARIMAX$(1,1,0)(2,1,0)_{168}$ model. MAE: 0.28, MSE: 0.20

Concerning the exogenous values, we used some almost independent features that were clearly correlated to the evolution of power consumption (as shown by the correlation matrix, Fig. 5): air temperature, downwelling IR flux, weekdays and weekends. they do not seem to offer a real improvement on the result given by SARIMAX. With the dates that we use here as an example, we see on the contrary that the results are a little bit worse when considering the exogenous values (Fig. 14).

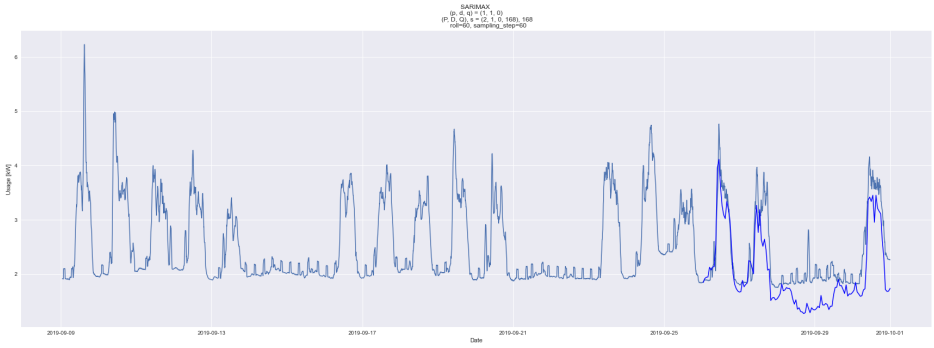

Figure 14: Forecasting with a SARIMAX$(1,1,0)(2,1,0)_{168}$ model with exogenes. MAE: 0.41, MSE: 0.32

4.2.4 Prophet

Trend Changepoints Real time series frequently have abrupt changes in their trajectories. Prophet will automatically detect these change points and will allow the trend to adapt appropriately. We give the user the possibility to modify the change point detection range (with the "changepoint_range" parameter) or to add some change points to manually defined extreme values (with the "add_changepoints" parameter). However, we recommend not to modify these parameters because it would give worse results.

The most impactful parameter is probably the "changepoint_prior_scale" (CPS) parameter. It determines the flexibility of the trend. If it is too small, the trend will underfit, but if it is too large, the trend will overfit. By default, its value is 0.05, but there is a great uncertainty about the outcome, so we obtained better results with values around 0.001 (Fig. 16) and we reduced significantly the uncertainty.

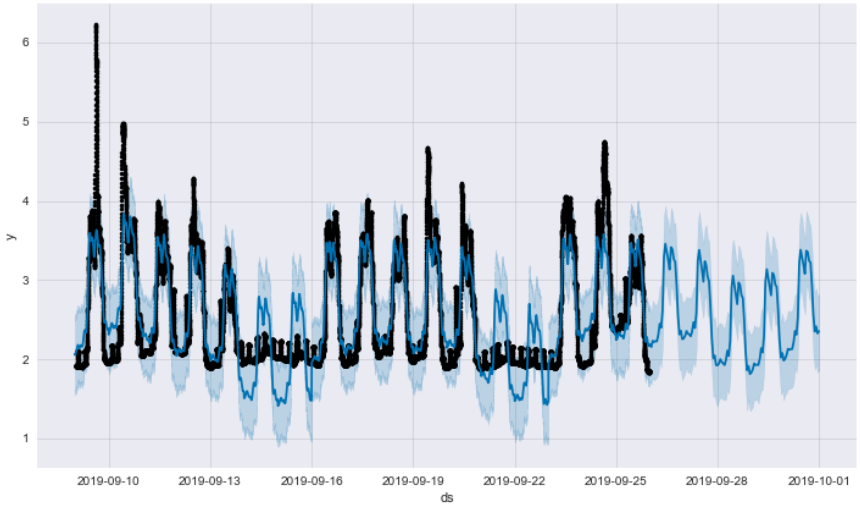

Figure 15: Forecasting with a Prophet model without weekend seasonality. MAE: 0.36, MSE: 0.25

Seasonalities The advantage of Prophet is the possibility to use several different seasonalities. We have used up to three possible seasonalities: a daily seasonality, a weekly seasonality and a seasonality based on the division of the week between weekdays and weekends (called "weekend seasonality"). It is clearly this last seasonality, which we have added by hand, that allows Prophet to correctly understand the articulation of the weekly power consumption between weekdays and weekends(Fig.15 and Fig. 16).

We did not use Prophet holidays parameters because our models are trained on only a few weeks (it is not yearly data but daily data). However, using the holidays parameters would fine-tune a little bit more our model.

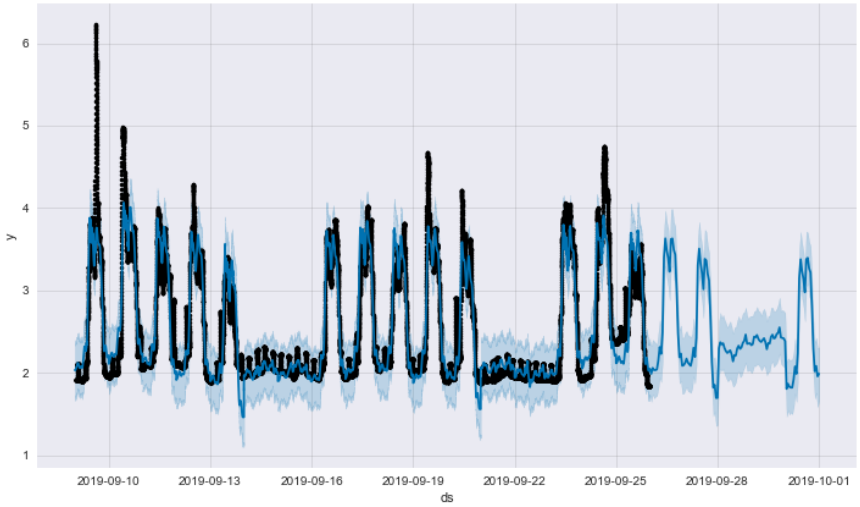

Figure 16: Forecasting with a Prophet model with weekend seasonality. MAE: 0.27, MSE: 0.18

Fourier order for seasonalities Seasonalities are calculated using a partial Fourier sum (eq.6). By default, order 3 is used for weekly seasonality, but we increased it to fit higher frequency changes, especially during a single day.

Exogenous values We can also use exogenous values with Prophet. Their use improves the results (in particular taking into account the air temperature) but with no drastic impact on them. However, Prophet runs very fast and we can easily add exogenous values without losing speed, unlike models like SARIMAX which becomes slower and slower as we add exogenous features.

4.3 Model comparison

Model	MAE	MSE
AR(24)	0.59	0.52
AR(96)	0.45	0.35
AR(96), exog.	**0.34**	**0.27**
ARIMA(1,1,0)	0.57	0.53
SARIMAX, s=24	0.65	0.71
SARIMAX, s=24, exog.	1.01	1.61
SARIMAX, s=168	**0.28**	**0.20**
SARIMAX, s=168, exog.	0.41	0.32
Prophet	0.36	0.25
Prophet, exog.	0.31	0.21
Prophet, exog., weekend seasonality	**0.27**	**0.18**

Table 1: Performances comparison. The parameters of SARIMAX are $(1, 1, 0)(2, 1, 0)$.

5 Further work

The fields of time series forecasting is vast, as is the project. Many other tracks can be explored to try to provide better predictions.

5.1 Other models

First, time series forecasting is a highly studied topic and other models exist, many of them described in F. Dama and C. Sinoquet's survey[5]. We have focused on ARIMA-like linear models and Prophet for their interpretability.

Non-Linear Models Polynomial autoregressive model PAR(q, p) has non-linear dependencies with past values: very similar to AR(p), it is a q-degree polynomial function of the p past values. It has an ability to capture memory effects.[6] Another variation of the AR model is the smooth transition autoregressive model, that allows transitions from an AR model to another.[7]

Deep Learning Neural networks are another type of non-linear models used in forecasting. Recurrent neural networks (RNNs) and long short-term memory networks (LSTMs), which are a subclass of RNNs are widely used in time series forecasting and natural language processing (NLP), two fields in which models have to handle sequential data, They have a memory component that keeps track of previous data on the long term. They have been used for solar power forecasting for instance.[8] A most recent deep learning framework, widely used in NLP is represented by Transformers[9] that have the specificity of taking as an input the entire sequence at once rather than element by element, and summarizes the information contained in a sequence in an embedding vector. They are able to capture the dependencies within the time series.

5.2 Feature engineering

We focused on predicting the Usage variable that includes the total power consumption of the DrahiX zone 2. However, as we have seen it in Fig.1, the greatest part of power consumption comes from the variable Plugs and HeatingCoolingTotal, while Ventilation, Lights and WaterHeater are less significant and have smaller variations.

- **Plugs**. The variation of plugs consumption is the most predicable and the most regular. It can be decomposed in two parts: an almost constant component of 1.9 kW and a daily seasonality with an amplitude of 0.3 kW, and that is almost equal to zero on weekends, as nobody is in the office. The pattern is clearly visible in Fig. 3. There is a consumption peak between March and April 2019 that we cannot understand with the available data, but for which we could ask the DrahiX occupants.
- **HeatingCoolingTotal**. There is a daily and weekly seasonality for this variable too, and a stronger correlation to AirTemp than Usage has.

Thus, we could separate Usage into HeatingCoolingTotal on the one hand, that we could predict more precisely using exogenous variables lire AirTemp, and the rest that seems less dependent on it.

References

[1] R. Nau, *Statistical Forecasting: Notes on Regression and Time Series Analysis*. Durham: Fuqua School of Business, Duke University, 2016. [Online]. Available: http://people.duke.edu/~rnau/411home.htm.

[2] R. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, English, 3rd. Australia: OTexts, 2021.

[3] S. Taylor and B. Letham, "Forecasting at scale," *PeerJ Preprints*, 2017.

[4] S. 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," in *2016 IEEE International Energy Conference (ENERGYCON)*, 2016, pp. 1–6. DOI: 10.1109/ENERGYCON.2016.7514029.

[5] F. Dama and C. Sinoquet, *Time series analysis and modeling to forecast: A survey*, 2021. arXiv: 2104.00164 [cs.LG].

[6] O. Karakuş, E. E. Kuruoğlu, and M. A. Altınkaya, "One-day ahead wind speed/power prediction based on polynomial autoregressive model," *IET Renewable Power Generation*, vol. 11, no. 11, pp. 1430–1439, 2017. DOI: https://doi.org/10.1049/iet-rpg.2016.0972. eprint: https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/iet-rpg.2016.0972. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-rpg.2016.0972.

[7] Ø. Eitrheim and T. Teräsvirta, "Testing the adequacy of smooth transition autoregressive models," *Journal of Econometrics*, vol. 74, no. 1, pp. 59–75, 1996. [Online]. Available: https://EconPapers.repec.org/RePEc:eee:econom:v:74:y:1996:i:1:p:59-75.

[8] A. Gensler, J. Henze, B. Sick, and N. Raabe, "Deep learning for solar power forecasting — an approach using autoencoder and lstm neural networks," *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 002 858–002 865, 2016.

[9] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.