Forside

Eksamensinformationer
N340016102 - SPDM801: Speciale i datalogi - Marcel Meimbresse - Juni 2021

Besvarelsen afleveres af
Marcel Meimbresse
mamei16@student.sdu.dk

Administration
Inge Huus
huus@imada.sdu.dk
+4565508958

Bedømmere
Arthur Zimek
Eksaminator
zimek@imada.sdu.dk
+4565509071

Panagiotis Karras
Censor
piekarras@gmail.com
+4591416469

Marco Chiarandini
Intern medbedømmer
marco@imada.sdu.dk
+4565504031

Vær opmærksom på titlen på specialet / Pay attention to the title of your thesis
Du skal være opmærksom på, at den titel du angiver, skal stemme overens med den titel som studienævnet har godkendt. Du skal angive titlen på både dansk og engelsk. Har dit speciale ikke en dansk titel, skal du anfører den engelske titel som dansk titel også.

Vær desuden ekstra opmærksom på stave- og slåfejl.

ENGLISH

Please note that the title you enter must correspond to the title approved by your Study Board. You must enter the title in both Danish and English. If your thesis does not have a Danish title, you must state the English title as a Danish title as well.

Also pay extra attention to spelling and typing errors.

Links

Filer

Besvarelsesinformationer

Titel: Normaliseret Elforbrug
Titel, engelsk: Normalized Electricity Consumption

Department of Mathematics
and Computer Science

University of Southern Denmark

Master Thesis in Computer Science

Normalized Electricity Consumption

Author
Marcel Meimbresse
Exam nr.
92472655
Email Address
mamei16@student.sdu.dk

Supervisor
Arthur Zimek
Co-supervisors
Marco Chiarandini
Mette Gamst (Energinet)

June 1, 2021

Abstract

English In the past, a multitude of research papers have been written on the subject of predicting electricity consumption in a wide variety of geographical locations and at different temporal and spatial resolutions. This thesis investigates how a selection of univariate and weather parameter based multivariate prediction methods can be used produce medium to long term forecasts of the net electricity consumption of Denmark, split geographically into the two price areas DK1 and DK2 by the Great Belt. Public raw weather data was provided by the Danish institute of meteorology, while public quality controlled electricity consumption data covering the years 2011-2020 was provided by Energinet. The raw weather data was cleaned and subjected to geospatial averaging and temporal interpolation to obtain a single gap-less time series for each weather parameter. For the purpose of prediction, the data was split into a training and a test set, consisting of 7 and 2 years, respectively. Two naive prediction models were included to form a baseline. All forecasts are given at a monthly time resolution. The results show that both regression-type machine learning models and the alternative Pattern Sequence based Forecasting (PSF) method achieved lower prediction errors on the test data set, which were calculated using the mean absolute error and mean absolute percentage error per month. The lowest prediction errors achieved by the naive baseline were 45 334.24 MWh/2.59 % and 27 901.35 MWh/2.61 % for DK1 and DK2, respectively. The overall lowest prediction error for DK1 (24458.57 MWh/1.47 %) was achieved by PSF, while a principal component regression model using integrated weather and monthly dummy variables achieved the lowest overall error for DK2 (21377.61 MWh/2.03 %). Due to the lack of a separate final test set, only limited conclusions can be drawn from the results. The results suggest that PSF, ARIMA and multiple linear regression models may be promising candidates for future research to determine if they can consistently outperform the naive prediction models currently in use.

Danish Der er tidligere blevet skrevet en lang række forskningsartikler om forudsigelsen af elforbruget i et bredt antal forskellige geografiske områder og med skiftende tidsmæssige og rumlige opløsninger. I dette speciale undersøges det, hvorvidt et udvalg af univariate og vejrparameterbaserede multivariate forudsigelsesmetoder kan anvendes til at producere mellem- og langsigtede prognoser for netto-elforbruget i Danmark, der geografisk er opdelt i de to budzoner DK1 og DK2 ved Storebælt. Åbne offentlige rå vejrdata blev tilgængeliggjort af Danmarks Meteorologiske Institut, mens offentlige kvalitetskontrollerede elforbrugsdata for årene 2011-2020 blev tilgængeliggjort af Energinet. De rå vejrdata blev renset og underkastet geospatial gennemsnitsberegning og tidsmæssig interpolation for at opnå en enkel tidsserie uden huller for hver vejrparameter. Med henblik på forudsigelse blev dataene opdelt i et trænings- og et testsæt bestående af henholdsvis 7 og 2 år. To naive forudsigelsesmodeller blev inddraget som udgangspunkt. Alle prognoser er givet med en månedlig tidsopløsning. Resultaterne viser, at både regressionsbaserede maskinlæringsmodeller og den alternative Pattern Sequence based Forecasting (PSF) metode opnåede lavere forudsigelsesfejl på testdatasættet, som blev beregnet ved hjælp af den gennemsnitlige absolutte fejl og den gennemsnitlige absolutte procentvise fejl pr. måned. De laveste forudsigelsesfejl der blev opnået med de naive modeller var 45 334.24 MWh/2,59 % og 27 901.35 MWh/2,61 % for henholdsvis DK1 og DK2. Den samlet set laveste forudsigelsesfejl for DK1 (24458,57 MWh/1,47 %) blev opnået med PSF, mens en hovedkomponentanalysebaseret regressionsmodel med integrerede vejr- og månedlige dummy-variabler opnåede den samlet set laveste fejl for DK2 (21377,61 MWh/2,03 %). På grund af manglen på et separat endeligt testsæt kan der kun drages begrænsede konklusioner af resultaterne. Resultaterne tyder på, at PSF, ARIMA og multiple lineære regressionsmodeller kan være lovende kandidater til fremtidig forskning for at afgøre, om de konsekvent kan overgå de naive forudsigelsesmodeller, der anvendes i øjeblikket.

Contents

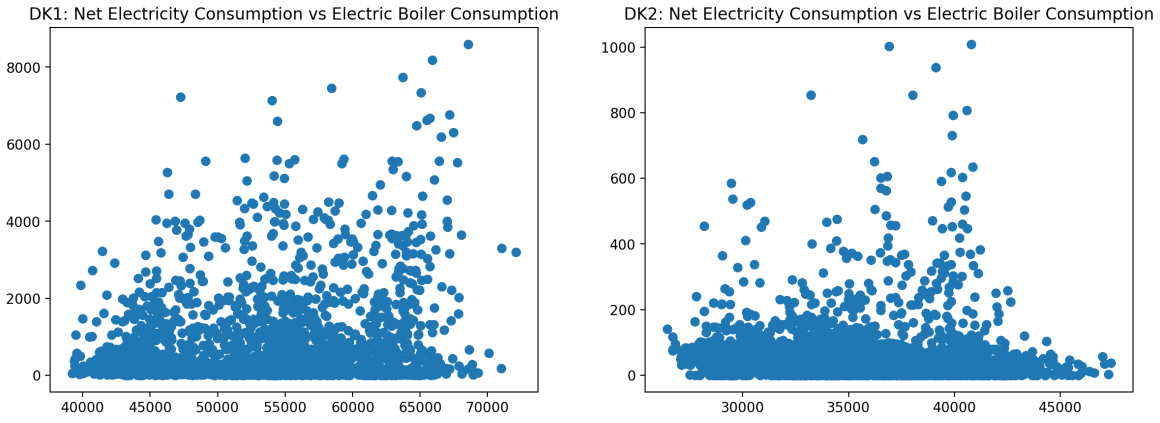
1 Introduction

The problem of predicting electricity consumption is of high interest for electricity providers and consumers alike, as it allows the former to adapt/expand their infrastructure and the latter to, for example, optimize their energy efficiency. As such, there exists considerable research on this prediction task at different spatial and temporal resolutions, from estimating consumption for a single building at a 15 minute prediction resolution [36] to predicting the total consumption of an entire country over a 1 year time frame [27]. For the purpose of this project, predicting the total electricity consumption for the whole of Denmark, split into the two electricity zones DK1 and DK2, at a medium term time interval is the main objective. A secondary objective is to investigate and possibly utilize the effect that weather parameters have on electricity consumption, by applying machine learning methods. The project was carried out in collaboration with Energinet, the Danish national transmission system operator for electricity and natural gas.

2 Data Collection

2.1 Electricity Consumption Data

The main electricity consumption data was supplied by Energinet and covers the energy consumption of the two electricity zones DK1 and DK2 of Denmark (where DK1 is west of the Great Belt and DK2 east of it), respectively, over a period of 9 years (2011-2020). It is assumed to be quality controlled and hence ready for use, without the need for data cleaning. However, this assumption only holds for data that has a certain age. Data that is roughly less than two weeks old may not have been verified yet, and as such may be erroneous. Therefore, this type of very recent data was not included in the data set. The target variable to be predicted for the purpose of this project is the net energy consumption, which excludes power line transmission loss, but by default includes the consumption of electric boilers. It is measured in Megawatt-hours (MWh). The relationship between electric boiler consumption and the net electricity consumption was analyzed. For this purpose, the former was first subtracted from the latter. Afterwards, scatter plots were generated and the linear correlation between net and electric boiler consumption was computed. To make the scatter plot clearer, both time series were first summed to a daily time resolution. The scatter plots shown in Figure 1 suggest that there is very little correlation between the two. This observation is supported by the computed Pearson correlation coefficients, which are 0.148 and -0.013 for DK1 and DK2, respectively.



(a) DK1 (b) DK2

Figure 1: Scatter plots showing the relationship between net electricity consumption (x-axis) and electric boiler consumption at a daily time resolution in the electricity zones DK1 and DK2.

Based on these findings, the decision was made to remove the electric boiler consumption from the net consumption data, since its near random behavior would be difficult to predict, thus leading to

inaccurate prediction errors when evaluating machine learning methods.

2.2 Weather Data

Raw weather observation data for Denmark has been made available to the public by the Danish Meteorological Institute (DMI) via an API in the year 2020, with quality controlled observations (called climate data) scheduled to be released in 2021. Since the data is provided raw, it has to be pre-processed before use in prediction models, in order to handle erroneous and missing measurements. Errors might be caused by measurement equipment failure, while missing data can also be caused by problems in transmission or blackouts.

While DMI offers 44 weather parameters through the API, only a subset was used. Choosing an appropriate subset can be done both empirically, based on previous scientific literature on the subject matter, and intuitively, as some parameters measure the same unit at different time resolutions. All temperature-related parameters, as well as global solar irradiance, wind speed and humidity were measured using a rolling average at a one hour time resolution. The precipitation parameter instead measured the total amount of rain accumulated over 1 hour, while the snow depth was measured manually once every day at a few select weather stations only. Sunshine duration and leaf moisture measured the number of minutes of sunshine and moist leaves in the past hour, respectively. Both visibility and pressure were originally measured at a 10 minute time resolution. To bring these two parameters into line with the others, a rolling average at an hourly time resolution was applied. Air temperature was measured 2 m above the terrain, grass temperature at 5-20 cm above the terrain and soil temperature at a depth of 10 cm below ground.

Besides the most commonly used parameter, temperature, which has been shown to provide useful prediction power in past studies [19], to the best of the student's knowledge the correlation between the others and energy consumption would yet have to be investigated for the region of Denmark. Exemplary parameters that were excluded are number of minutes with precipitation in the latest hour (since accumulated precipitation is already included), wind maximum, minimum and gusts (since brief changes in wind will hardly influence energy consumption), wind direction, cloud height, pressure at sea level, quarters of the earth covered by snow (since its unit is unspecified), weather (since its unit is an enumeration) and cloud cover (because it can be thought of as the opposite of sun shine duration and is explained implicitly by solar radiation). Since the parameters will later be used to make predictions at an hourly to monthly time resolution, including a parameter at a 10 minute resolution when it also available at an hourly time resolution was deemed redundant. In total, 12 parameters were included (See Table 1).

Table 1: Table of all included weather parameters, together with their description, measurement unit and original measurement frequency. Source: [2].

Name	Unit	Description	Data update frequency
temp_mean_past1h	°C	Latest hour's mean air temperature measured 2 m over terrain	Hourly
temp_grass_mean_past1h	°C	Latest hour's mean air temperature measured at grass height (5-20 cm over terrain)	Hourly
temp_soil_mean_past1h	°C	Latest hour's mean temperature measured at a depth of 10 cm	Hourly
radia_glob_past1h	W/m^2	Mean intensity of global radiation in the latest hour	Hourly
sun_last1h_glob	minutes	Number of minutes with sunshine the latest hour	Hourly
leav_hum_dur_past1h	minutes	Number of minutes with leaf moisture the latest hour	Hourly
wind_speed_past1h	m/s	Latest hour's mean wind speed measured 10 m over terrain	Hourly
visibility	m	Present visibility	10 min
precip_past1h	kg/m^2	Accumulated precipitation in the latest hour or the code -0,1, which means "traces of precipitation, less than 0.1 kg/m^2". kg/m^2 is equivalent to mm. (see Codes (metObs))	Hourly
pressure	hPa	Atmospheric pressure at station level	10 min
snow_depth_man	cm	Snow depth (measured manually) or the code -1, which means "less than 0.5 cm" (see Codes (metObs))	Daily
humidity_past1h	%	Latest hour's mean for relative humidity measured 2 m over terrain	Hourly

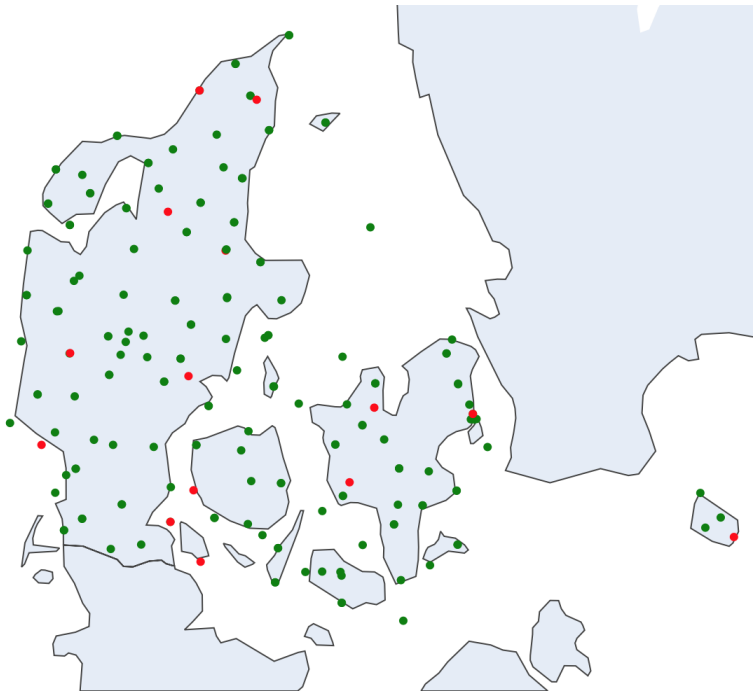


Figure 2: Locations of the 184 weather stations in Denmark. Stations colored red were found to be error-prone for a certain parameter. Interactive map available at [5]. Note that points may overlap.

3 Data Cleaning

3.1 Untrustworthy Stations

Because the data was expected to contain erroneous measurements and was collected by 184 weather stations distributed over Denmark, it was first analyzed which stations are responsible for the top and bottom 100 most extreme measurements. The motivation for this was to determine if there are certain stations that make out a large proportion of extreme measurements for a given parameter, in which case they could be deemed untrustworthy and hence all their measurements for said parameter excluded from the data set. In the later stage of data cleaning, when out-of-bounds values were replaced by "Not a Number" (Nan) values, it was additionally investigated which stations were responsible for the most out-of-bounds values. The set of stations found via this method matched that found via the initial process. Through this analysis, a total of 13 stations were found to have a disproportionately large share in the 200 most extreme measurements and were excluded for their respective parameter. Figure 2 shows the location of all stations in Denmark and highlights the 13 untrustworthy ones.

3.2 Code Removal

Some of the weather parameters use codes to represent specific weather situations. However, these codes are placed among the actual weather measurements, instead of a separate data column, and hence need to removed to obtain valid measurements. Only 2 of the 12 parameters, namely "snow_depth_man" and "precip_past1h" are affected by this. Respectively, negative values of -1 and -0.1 are used to represent very low amounts of rain or snow (see Table 1). Hence, these codes were replaced by the value 0, since this is the closest valid value in the respective units of both parameters.

3.3 Extrema Removal

To determine the extent of erroneous data, the raw weather observations for each parameter were plotted in time. Visual inspection was facilitated by inserting two horizontal lines in each plot, repre-

senting the upper and lower bound of each parameter, respectively. These bounds were established by consulting public DMI records of historic extremes for each parameter [3] or, in the cases of precipitation, solar radiation, visibility, pressure and snow depth, estimated based on more general publicly available data. More specifically, the upper bound for precipitation was set to be 60 mm rain, twice that of a so-called "double downpour", which is defined by DMI to be more than 30 mm precipitation over a period of 30 minutes. The upper bound for solar radiation was estimated more roughly to be 1120 W/m^2, since sources defining normal and extreme solar radiation [6] are scarce. Extrema for visibility and pressure were sourced from Wikipedia [10] [9], whereas the upper bound for snow was sourced from a TV2 news article [8].

All measurements that were out of bounds, as determined by the upper and lower bounds established before, were replaced with NaN values. In total, only 2832 out of 32 238 688 measurements were out of bounds, corresponding to 0.009 %. Afterwards, all NaN values that were surrounded by valid values were linearly interpolated from their preceding and succeeding values (neighboring in time). The missing values were interpolated for each weather station separately.

3.4 Outlier Detection and Removal

Following the removal and re-estimation of out-of-bounds values, another method for removing remaining erroneous values, which manifested as outliers in the time series plots, was needed. To this end, the two-sided median method, a simple outlier detection method proposed by Basu and Meckesheimer in 2007 [13] was used. For any data point x_t at time step t, the median of the $2k$ preceding and succeeding values is calculated. In the original paper, the current data point is not included in this calculation, but due to inefficiencies in the *Pandas* software library used for this project, excluding it was not possible. Hence, for every data point x_t, the median $m_t(x_{t-k}, ..., x_{t-1}, x_t, x_{t+1}, ..., x_{t+k})$ is computed, where m_t denotes the median. Thereafter, x_t is classified as an outlier if and only if $|x_t - m_t| \geq \tau$, where τ represents a specified threshold. Both the neighbourhood size k and the threshold τ have to be experimentally determined, since their effect depends on the time series data at hand and may even change over the course of the time series itself. In [13], it is mentioned than "one might also use information from the actual signal or process to determine appropriate values for window width and threshold (for example, the percent deviation from the mean signal)". In this project, the threshold was defined as $c\sigma$, where σ is the standard deviation of the given parameter measurements and $c \in \mathbb{R}^+$ is a coefficient that, in combination with k, is chosen such that only a fraction of all data is classified as outlying, while also correctly detecting most outliers that were previously discovered via visual inspection. Figure 3 showcases the two-sided median method, applied to a small sample of data from January 2017, measured by a single weather station and the parameter temp_mean_past1h. The sample was chosen by picking the first weather station to appear in the dataset, whereafter the first instance of visible outliers was isolated. The neighbourhood size k was set to 17, while the threshold τ was chosen to be one standard deviation from the mean of temp_mean_past1h. The figure shows that most if not all outliers found in this particular sample were detected by the outlier detection method, with a single suspicious measurement on the 8th of January remaining. This result aligns with the proposed goal of choosing k and τ such that most outliers will be detected, while only removing a minimal amount of valid data. Table 2 contains the hyperparameters chosen for each weather parameter, as well as the amount of data flagged as outlying.

The results of cleaning the raw data is illustrated in Figures 4 and 5, showing the data before and after it was cleaned, for the two select parameters "temp_mean_past1h" and "radia_glob_past1h", which measure air temperature and global solar irradiance, respectively. The before/after plots for the other parameters can be found in the appendix.

Table 2: Window size k and coefficient c (in threshold $\tau = c\sigma$) of the two-sided median outlier detection method chosen for each weather parameter, as well as the percentage of data that was detected as outlying.

Parameter	k	c	% of data flagged as outlying
temp_mean_past1h	17	1	0.021
temp_grass_mean_past1h	12	1	0.017
temp_soil_mean_past1h	3	1	0
radia_glob_past1h	7	1	0.76
sun_last1h_glob	3	2	0.075
leav_hum_dur_past1h	3	2	0.06
wind_speed_past1h	3	1	0.02
visibility	3	1.5	0.045
precip_past1h	3	8	0.089
pressure	10	1	0.001
snow_depth_man	3	3	0.039
humidity_past1h	3	1	0.017

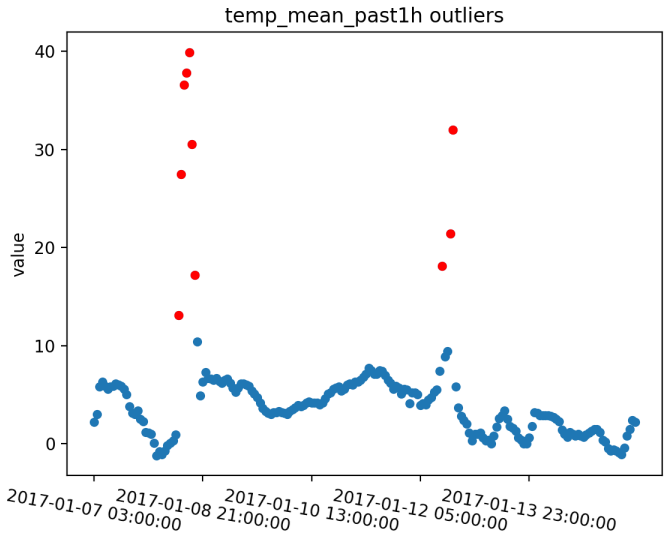
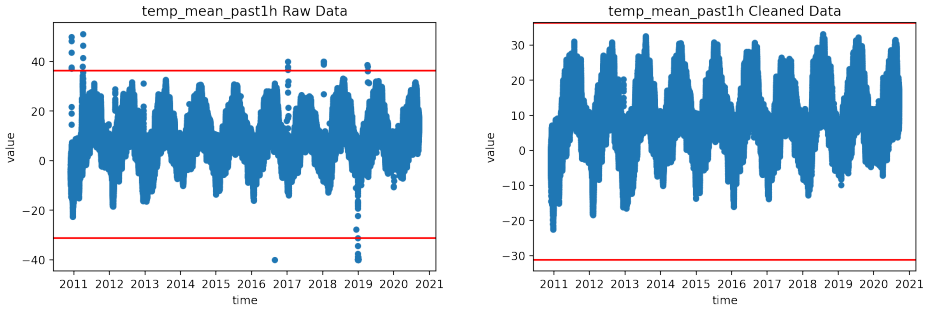


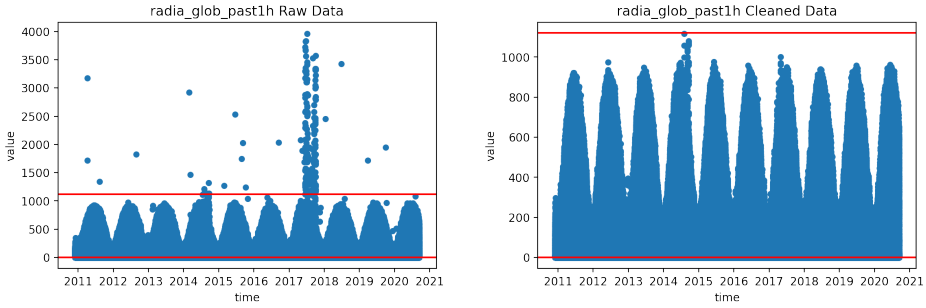
Figure 3: The two-sided median outlier detection method (with $k = 17$ and $\tau = 1\sigma$) applied to a small section of uncleaned weather observation data measured in the parameter "temp_mean_past1h" by a single weather station. Detected outliers are marked red. The date labels on the x-axis are written in the format year-month-day.



(a) Raw Temperature Data (b) Cleaned Temperature Data

Figure 4: Temperature measurements before/after cleaning (Red lines show upper/lower bound for value). The y-axis of Plot (b) was automatically adjusted due to the removal of extreme values (to minimize dead space in the plot)

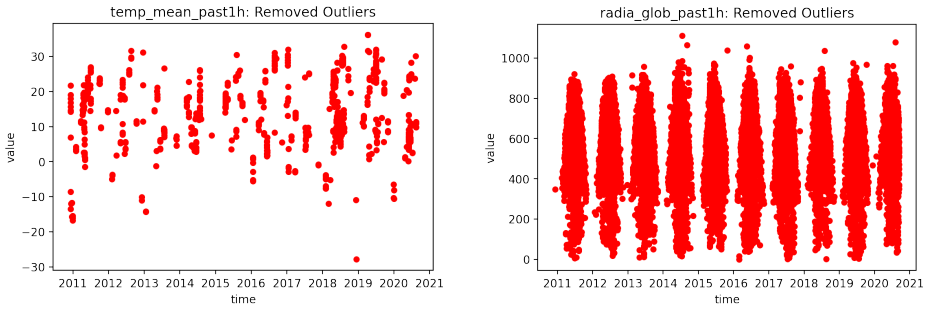
.



(a) Raw Solar Irradiance Data (b) Cleaned Solar Irradiance Data

Figure 5: Solar irradiance measurements before/after cleaning (Red lines show upper/lower bound of the parameter unit)

As evident in plots (a) and (b) of figures 4 and 5, not all outliers and spurious data points could be removed via the applied methods. For example, a vertical row of points can be seen in the cleaned temperature plot roughly above the 2013 tick on the x-axis. Similar vertical rows of points that are likely to be erroneous are evident in the solar irradiance cleaned data plot between 2014 and 2015, as well as between 2017 and 2018. Based on visual inspection only, the amount of these outliers appears to be minute when compared to the total amount of measurements. However, depending on the robustness of a prediction model, they might still negatively influence it.



(a) Air temperature outliers (b) Solar irradiance outliers

Figure 6: Air temperature and Solar irradiance outliers that were removed from the data set and subsequently interpolated from valid data.

While the two-sided median outlier removal method that was applied to clean the data was unable to remove all outliers, it may have also removed a portion of valid data. This is illustrated by plots a and b of Figure 6, which showcases the detected outliers that subsequently were removed from the data set. Although the amount of removed data may seem drastic based on the plots, especially in the case of the solar irradiance data, only 0.021 % of air temperature and 0.76 % of solar irradiance data was discarded, respectively. In addition, all removed data points were later re-estimated via temporal interpolation, hence why it was deemed less problematic to remove all visibly extreme outliers at the cost of removing a larger portion of valid data than the opposite.

4 Data Preparation

4.1 Interpolation

Even after the data has been cleaned, it may still not ready to be used in a machine learning model. On top of stations making erroneous measurements, they also sometimes do not measure anything at all, or at least their measurements are never returned to the central data collection location. This leads to gaps in the time series, which can reach from single hour gaps to several years of missing data in the case of certain weather stations. Depending on the size of the gap, different interpolation techniques may be applied to re-estimate the missing values.

In this project, the measurements of all stations in a price area were combined into a single time series before being used in machine learning models (See section 4.3). Therefore, it was necessary to choose when to apply interpolation: Either, it is applied to the measurements of each station before combining the measurements of all stations, or it only applied after this combination has been performed. In the latter case, only small gaps will remain in the time series, due to the amount of different stations and the fact that gaps rarely overlap, except when large outages in the communication networks occur.

4.1.1 Temporal Interpolation

When dealing with small gaps of missing data in time series, linear temporal interpolation can be applied to fill them from the surrounding valid data. More precisely, a missing value $\mu(\theta)_t$ of parameter θ at time step t can be interpolated as:

$$\mu(\theta)_t = (\mu(\theta)_{t-1} + \mu(\theta)_{t+1})/2 \tag{1}$$

Because all weather parameters were measured at an hourly time resolution, the average of the preceding and succeeding value is often a fairly close reproduction of what the original value would have been.

4.1.2 Geospatial Interpolation

While temporal interpolation can be useful for filling small gaps in time series data (for which it was used in this project), applying it to re-estimate data over longer time intervals can result in large errors, especially if the data is stochastic and is sampled at a high temporal resolution, as is the case with weather measurements. Since some weather stations have long intervals of missing data for certain parameters, another means of interpolation is required, if one wishes to apply it at a per-station level. To this end, the relatively high weather station density found in Denmark makes geospatial interpolation a feasible alternative. Here, the data from one or more neighbouring stations are used to interpolate missing data of another station. A trade-off may be needed to decide on the number of neighbouring stations from which to interpolate a missing value. Simply copying the value of the closest neighbour may not be best choice, since local weather phenomena such as rain or fog can occur at that station while not occurring at the target station. The measurement of the closest neighbour may also be erroneous. In both cases, subsequent applications of spatial interpolation at different stations could propagate a value to several, potentially faraway stations, which would be highly undesirable.

Therefore, it is necessary to incorporate the measurements of multiple stations in the interpolation, e.g., by computing their mean. At the other extreme, incorporating too many stations could lead to inaccurate re-estimations, since far away stations could be included when the target station is located in the least station-dense areas of Denmark. While a distance based weighted average of neighbouring stations could negate this drawback, it could also revive the drawback of using only a single station, if two stations are very close to each other.

In an effort to find the golden mean, the final spatial interpolation method is computed as the average of the two closest neighboring stations, with a fallback to copying a single value if only a single station in all of Denmark has recorded a value at the missing timestamp. This way, the propagation of a single value can only occur as a last measure, while it is otherwise minimized by calculating averages. Likewise, by using only the two closest stations, the issue of using the measurements of faraway stations is reduced.

Let $S(\theta, t) = \{s_1, s_2, ..., s_n\}$ be the set of all stations that have recorded a measurement of parameter θ at time t. A missing value $\mu(\theta)_{st}$ of parameter θ at station $s \in S(\theta, t)$ and time t can then be interpolated via the following computations:

$$D(s, \theta, t) = \{s_1, s_2, ..., s_{n-1} \in S(\theta, t) \setminus s \mid gc\_dist(s, s_i) \leq gc\_dist(s, s_j) \Leftrightarrow i < j\} \tag{2}$$

$$\mu(\theta)_{st} = \begin{cases} (\mu(\theta)_{D(s,\theta,t)_1} + \mu(\theta)_{D(s,\theta,t)_2})/2 & \text{if } |D(s, \theta, t)| \geq 2 \\ \mu(\theta)_{D(s,\theta,t)_1} & \text{otherwise} \end{cases} \tag{3}$$

Where $gc\_dist(x, y) = 2 \arcsin[\sqrt{\sin^2((x_1 - y_1)/2) + \cos(x_1)\cos(y_1)\sin^2((x_2 - y_2)/2)}]$, i.e., the distance between two stations $x = (x_1, x_2)$ and $y = (y_1, y_2)$ is calculated as the great circle distance between their coordinates, which are given as latitude and longitude (in that order) converted to radians. In equation 2, we define a set $D(s, \theta, t)$ of all stations except s that have measured parameter θ at time step t, which is ordered ascendingly by the distance between each station and station s. Note that we do not consider the case where all stations are missing an observation at a certain time step, since this would make spatial interpolation infeasible and is hence not relevant for this section. However, in practice, spatial interpolation was always succeeded by applying temporal interpolation, precisely to fill in those few time steps that are missing at all stations.

4.2 Temporal Downsampling

The process of lowering the original hourly temporal resolution of the weather and electricity consumption data to daily or monthly is known as temporal downsampling. A number of different reasons as to why this may be desirable exist, depending on the application area. In the specific case of weather variables, the relationship between them and electricity consumption appears to become more linear as the resolution is lowered, making linear models more applicable (See Section 6.2.1). Additionally, weather forecasts are not provided through a public API or even at all in the same format that the raw weather observations provided by DMI are in, making out-of-sample predictions difficult. It is possible to make so-called ex-post forecasts, in which future weather observations are known at the time of forecast. But these forecasts mainly serve the purpose of theoretical model research, as they have no real-world application. In real-life scenarios, one is instead forced to make so-called ex-ante forecasts, in which the future observations are unknown, necessitating some means of forecasting the external/independent weather variables, before they in turn can be used to forecast the target/dependent variable. These forecasts of external variables are often made by applying popular univariate time series prediction methods such as ARIMA or even LSTM models. However, since even complex commercial weather models struggle to produce accurate daily or even hourly forecasts months into the future, the above-mentioned univariate models are unlikely to produce usable forecasts at such time resolutions. Therefore, by applying temporal downsampling, univariate forecasting methods may have a better chance to produce usable weather forecasts, since short-lived and hard to predict weather phenomena are smoothed out.

Among the 12 weather parameters considered in this project, two require special attention when applying temporal downsampling, since they are accumulative. This means, that they need to be summed,

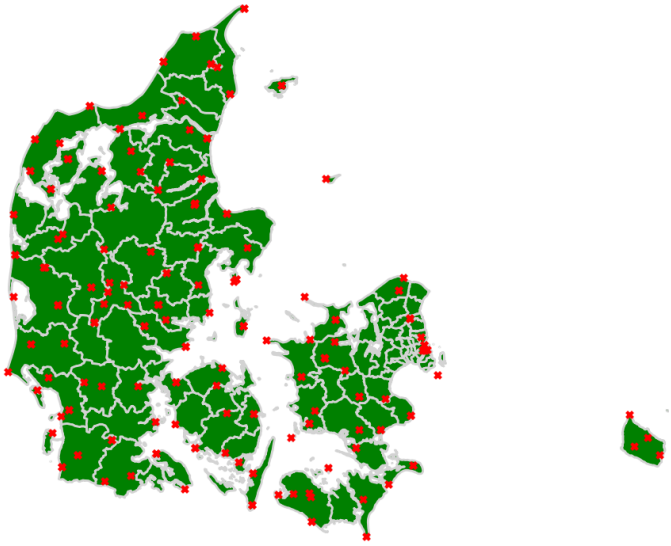


Figure 7: Locations of the 184 weather stations in their respective Danish municipalities. Municipal boundaries are coloured light grey.

while the rest need to be averaged to obtain meaningful lower-resolution results. The two accumulative parameters are sunshine duration (sun_last1h_glob) and leaf humidity duration (leav_hum_dur_past1h), both of which are measured in minutes.

4.3 Geospatial Averaging

In the following sections, three strategies to obtain a weighted sum of weather observations from multiple stations and for a single weather parameter will be presented. The weight of a station will be denoted $w_s(\theta)$, where s is a station in the set of stations S and θ is a weather parameter. To emphasize that the different methods entail different weight values, an apostrophe is added to the weight notation for each new method, i.e., $w'_s(\theta)$ and $w''_s(\theta)$ for the second and third weight calculation formulae, respectively. Given that the aim is to calculate a geospatial average, the sum of all weights is made to equal 1.

4.3.1 Municipality Based

The models in sections 6.2.4 and 6.2.5 using weather parameters are based on simple geographical averaging to obtain a single parameter observation for every time step t. This average is taken over all weather stations in DK1 and DK2, respectively. However, through this approach, each station is given the same weight in the calculation, regardless of whether a station is located in a large city or on a small island several kilometres from the coast. Since the energy consumption of the city is likely far higher than that of the island, it seems logical to attribute more importance to the observations of the station in the city than to those of the island station, given that weather influences the amount of electricity consumed. At the time of this project, the public Energinet dataset with the highest available spatial resolution provides the monthly electricity consumption for each municipality in Denmark, from the year 2013 to 2020. It appears that this data set has not been quality controlled or does not measure the total energy consumption of each municipality, since the total monthly consumption of all municipalities does not equal the total consumption of Denmark, as provided by the main Energinet dataset used in this project. Therefore, it cannot be used directly in machine learning models as a prediction target. Nonetheless, it may still be useful to establish the relative electricity consumption of each municipality, after which this information can be used to assign a weight to each weather station and compute a weighted geographical average. Figure 7 shows how Denmark in subdivided into municipalities and the location of all weather stations.

Let S be the set of weather stations, M the set of Municipalities and $m(s)$ the municipality $m \in M$ that

contains station $s \in S$ geographically. Further, denote by $\bar{c}_m$ the mean monthly electricity consumption of $m \in M$ and let $S_m \subset S$ be the set of all stations that are geographically located in municipality m. Then the weight $w_s(\theta)$ of each station $s \in S$ can be calculated in the following way:

$$x_{m(s)} = \frac{\bar{c}_{m(s)}}{\sum_{m' \in M} \bar{c}_{m'}} \tag{4}$$

$$w_s(\theta) = \frac{x_m(s)}{|S_{m(s)}|} \tag{5}$$

As evident in equation 5, this approach will distribute the relative electricity consumption of each municipality $m \in M$ uniformly among all stations $S_{m(s)}$ that it contains.

However, there exist factors that suggest using a different way of distributing the consumption might be preferable. For example, a municipality such a Frederikshavn, situated in the very north of Denmark, contains both a station at the Skagen lighthouse on the northern tip of the country and one in Voerså, which is located on the east coast 60 km away from the first station. This distance and the difference in siting may result in weather conditions being significantly different between the stations. Even for two stations that are more closely situated, one may consistently produce inaccurate measurements due to, e.g., invalid calibration, which would not be caught by an outlier detection mechanism. It is therefore possible that some stations may be more "representative" of the overall weather in their respective municipality than others and hence their observations more important for predicting electricity consumption. To try and capture the level of representativity of each station, one could calculate the correlation between each station's observations and the electricity consumption of the municipality it is located in. A station that achieves a higher correlation compared to its municipal neighbours would then be attributed a bigger share of relative electricity consumption than the others and hence a bigger weight in the computation of the weighed average across all stations in an electricity zone. Let $\vec{c}_m$ be the time series of monthly electricity consumption of municipality m and $\vec{\mu}_s(\theta)$ the time series of observations of a single parameter θ at station s, which have been averaged or summed to a monthly time resolution to match the temporal resolution of the municipal consumption data. In addition, denote by $corr(\vec{\mu}_s(\theta), \vec{c}_m)$ the linear correlation between $\vec{\mu}_s(\theta)$ and $\vec{c}_m$. Then, another way to calculate the weight of a station $s \in S$ is:

$$z_{sm(s)}(\theta) = \frac{corr(\vec{\mu}_s(\theta), \vec{c}_{m(s)})}{\sum_{s' \in S_m} corr(\vec{\mu}_{s'}(\theta), \vec{c}_{m(s)})} \tag{6}$$

$$w'_s(\theta) = z_{sm(s)} x_{m(s)} \tag{7}$$

With equation 4 remaining unchanged.

4.3.2 Voronoi Tessellation Based

Using the existing municipal division of Denmark as a basis for geospatial averaging is straightforward, due to the publicly available geographical data facilitating the assignment of each station to its encompassing municipality and Energinet providing the electricity consumption of each municipality. Nevertheless, the weather stations are often located close to the boundary of their municipality (see Figure 7) and as such may not provide a good representation of the entire geographical area in terms of weather conditions. In fact, a station bordering a neighbouring municipality may be more representative of that area than of its own municipality. At the same time, as evident in Figure 7, a number of municipalities, mainly situated on the island of Zealand, do not contain any stations at all and are thus excluded from any municipality-based averaging methods. To tackle these issues, a method for re-partitioning Denmark based on the location of the different weather stations is needed. It should compute a partition such that each new region contains only a single station and each point in said region is closer to its assigned station than to any other, thus theoretically maximizing the representativity or coverage of each weather station.

Fortunately, such a method already exists and the resulting partition is known as a Dirichlet–Voronoi diagram/tesselation/partition or Thiessen polygons. More specifically, given a set of points $P = \{p_1, p_2, ..., p_n\}$ in the Eucledian plane $\mathbb{E}^2$, a Voronoi Diagram $Vor(P)$ partitions $\mathbb{E}^2$ into subsets $V(p_i)$ such that $\bigcup_i V(p_i) = \mathbb{E}^2$ and an arbitrary point x on the plane is in $V(p_i)$ iff the Euclidean distance to p_i is smaller than to any other point $p_j, i \neq j$, i.e.,

$$V(p_i) = \{q \in \mathbb{E}^2 \mid dist(p_i, q) \leq dist(p_j, q)\} \qquad p_i, p_j \in P, i \neq j \tag{8}$$

, where $dist(x, y) = ||\vec{x} - \vec{y}||$. The concept of Voronoi diagrams generalizes to higher dimensions, but given that the weather stations' coordinates are two-dimensional, we restrict ourselves to this case.

Each subset $V(p_i)$ is called a Voronoi region or cell and only contains a single weather station, which is called a seed.

Note that this definition assumes that the points lay on the Euclidean plane, which is not the case for the stations, whose coordinates are given in longitude and latitude. To mitigate this problem, all coordinates were transformed to a zoned coordinate reference system covering all of Europe, for which Euclidean distance measurements are more accurate. Despite this, a margin of error remains and one should keep in mind that a two-dimensional Voronoi diagram computed from geographic data is only an approximate solution.

The Voronoi diagram is computed using a method from the *geovoronoi* library [4], which is based on *SciPy*'s `Voronoi` method [7]. The latter uses the *Qhull* library, written in the C programming language, to compute the diagram. In short, the computation proceeds as follows: Each point $q = (x, y)$ in the plane is "lifted" up onto a three-dimensional circular paraboloid by adding a third dimension to its coordinate such that the new $q' = (x, y, x^2 + y^2)$. Thereafter, the tangent plane of each lifted seed on the paraboloid is computed. Finally, the tangent planes are projected onto the original two-dimensional plane to form the Voronoi diagram, with the intersection lines between the planes forming the boundaries of the Voronoi regions. For a more detailed description of the mathematical background of this process, see Gallier [18].

Since the boundaries of Voronoi cells are convex polygons in two dimensions, *geovoronoi* works in conjunction with the *GeoPandas* library, which is used for all geographical data in this project and supports both polygonal and multi-polygonal objects. Denmark and all its municipalities are represented by multipolygons after reading the corresponding data into a `GeoDataFrame`. Since the Voronoi regions of seeds belonging to the boundary of the convex hull of P are unbounded, the *geovoronoi* library can use the multipolygon representing Denmark to bound these regions, such that they become bounded multipolygons. A multipolygon is the union of a set of non-intersecting polygons that is seen as a single geometric object. It may have holes and its polygons may be disjoint. Hence, a Voronoi region whose seed is located close to shore may extend over the water onto the opposite shore.

Figure 8 illustrates how the islands of electricity zone DK2 are re-partitioned based on the Voronoi Diagram by superimposing the bounded Voronoi cells onto the existing municipalities. Note how some of the cells extend across water. This image also showcases another benefit that may further increase the coverage of each station: Because some municipalities contain no stations at all, they were excluded from the computation of the municipality-based spatial average. Now, however, every municipality is covered by a station and thus the calculated weight (or importance) of each station may be more accurate. For example, a station bordering and covering a neighboring municipality with a high energy consumption while its own encompassing municipality has a low consumption will now be weighted based on the total consumption of both municipalities, as opposed to only its own (assuming that it also covers its encompassing municipality).

Let $a(m)$ be the multipolygon of municipality $m \in M$ and $V'(p_s)$ the bounded multipolygon based on the potentially unbounded Voronoi cell $V(p_s)$ such that $\cup_m a(m) = \cup_s V'(p_s)$. Then, the weight of each station $s \in S$ is calculated in the following manner:

$$w''_s(\theta) = \frac{\sum_{m \in M} \bar{c}_m (V'(p_s) \cap a(m))}{\sum_{m \in M} \bar{c}_m} \tag{9}$$

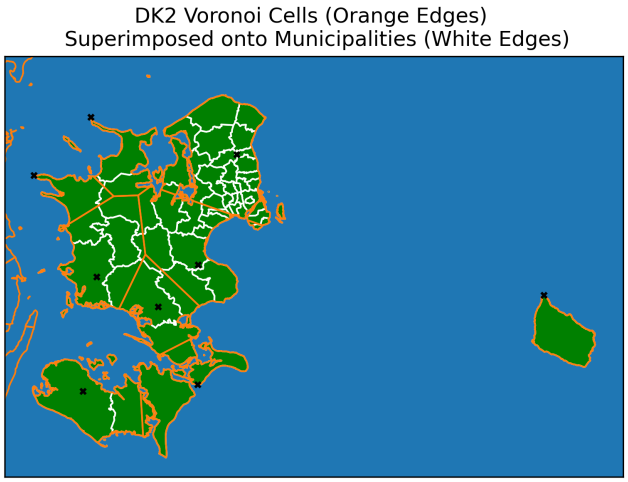


Figure 8: Voronoi Diagram based regions of electricity zone DK2 superimposed onto municipalities. Black crosses represent weather stations that measure temperature (among other parameters), which were used as seeds for the diagram.

5 Error Metrics

Whenever machine learning models are to be applied to a specific task, it is important to choose one or more meaningful error metrics that allow a researcher, developer or even a domain expert to interpret the forecasting performance of each model and compare it to other existing models. Given that the domain in this case is electricity consumption, it seems logical to include the Mean Absolute Error (MAE) as one of the error metrics, which is calculated in the following way:

$$MAE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|, \tag{10}$$

where y is the observed series and $\hat{y}$ is the predicted series. It can be easily interpreted and can be used directly to calculate the impact that the inaccuracy of a given model would have in terms of, e.g., expected electricity grid load or the trade of electricity on the market. However, a drawback of the MAE is that it cannot be used directly to compare models, for example, across different domains or even just across regions of different energy consumption. The fact that one of the key aspects of this project is that Denmark is split into the two electricity zones DK1 and DK2, with the former's electricity consumption being overall higher than the latter's, already justifies the inclusion of an error metric that enables this form of model comparison. In addition, choosing a metric that is used widely in literature, especially in the domain of electricity consumption, simplifies the comparison of the models implemented in this project to previous ones and thus also provides a form of external benchmark. The metric that best meets these criteria is the Mean Absolute Percentage Error (MAPE):

$$MAPE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{|y_i|} \tag{11}$$

Although it is useful for both intra- and inter-domain model comparison by converting the error to a percentage, it has multiple shortcomings that make it unsuitable as a primary error metric. Obviously, it cannot be applied to any series that contain zeroes, as division by zero is undefined, although some implementations attempt to mitigate this issue by replacing all zeroes by a small constant. Furthermore, while the metric has an upper bound of 1 (or 100 %) when it comes to predicted values smaller than the observed ones, it is unbounded w.r.t. overpredictions. This means that the MAPE

will favour a model that tends to underpredict over another that tends to overpredict, even if the former, in terms of MAE, is much less accurate. It is therefore highly advisable to always use the MAPE in conjunction with either another unbiased error metric such as the MAE or use forecast plots that show both y and $\hat{y}$, so as to determine if a model systematically underpredicts.

Due to these issues, the MAPE is only used to supplement the MAE in this project and is neither used for model selection nor hyperparameter tuning.

6 Model Descriptions

Before applying the different models, the data was split into training and testing data sets along the time axis, with the training set making up 80% of all data.

6.1 Univariate Models

6.1.1 Naive Baseline Models

In order to establish a baseline or benchmark for the later, more complex models to beat, two naive models were implemented, which forecast electricity consumption at a monthly temporal resolution. The first model is based on the assumption that any given year will be most similar to the one that came before it, in terms of electricity consumption. Thus, it simply repeats last years electricity consumption as the forecast for the current year. Therefore, to obtain forecasts for the validation set, consisting of two years, the naive model was allowed to use data from second to last year to forecast the last year.

The second model, on the other hand, is based on the assumption that the electricity consumption has not changed significantly over the past nine years. Hence, it computes the average year over all years in the training data and uses this as the prediction for all future years.

6.1.2 Autoregressive Integrated Moving Average (ARIMA)

Simple- and multiple regression models use one or more independent variables to estimate one or more dependent variables. However, another kind of regression model especially suited for univariate time series data uses only a single variable to predict its own value at a later time step. The simplest models of this kind have the form:

$$y_t = c + \phi y_{t-1} + \epsilon_t \tag{12}$$

Where c is a constant and the value y_t at the current time step is predicted by multiplying the value at the previous time step by an estimated coefficient ϕ. Such a model is known as a first order auto-regressive model, often abbreviated as AR(1). Though the error term ϵ, sometimes also known as the disturbance or random shock, is an un-observable random variable that is assumed to be independent identically distributed, where each ϵ_t is sampled from a normal distribution with zero mean and constant variance, we can estimate its value after computing equation 12 as:

$$\hat{\epsilon}_t = y_t - \hat{y}_t \tag{13}$$

If the error terms are autocorrelated, we may want to employ a model that uses this information. The simplest model of this kind has the form:

$$y_t = c + \theta \epsilon_{t-1} + \epsilon_t \tag{14}$$

This is known as a moving average model. Since only the error term of the previous time step is used, it has order 1 and is abbreviated as MA(1).

Some time series may by default not meet the stationarity criterion needed to produce good results with an ARIMA model. A stationary series is defined as a series with constant mean and variance. While a lack of constant variance can be mitigated by applying a power transformation such as the natural logarithm or the square root, a non-constant mean may be resolved by applying differencing to the series. The difference of a series is computed as:

$$w_t = y_t - y_{t-1} \tag{15}$$

It may be necessary to repeat this operation to reach stationarity. The order of differencing d is the number of iterations of operation (15). Hence, w_t is the first-order difference of y_t. The d-order difference of y_t can be obtained by computing:

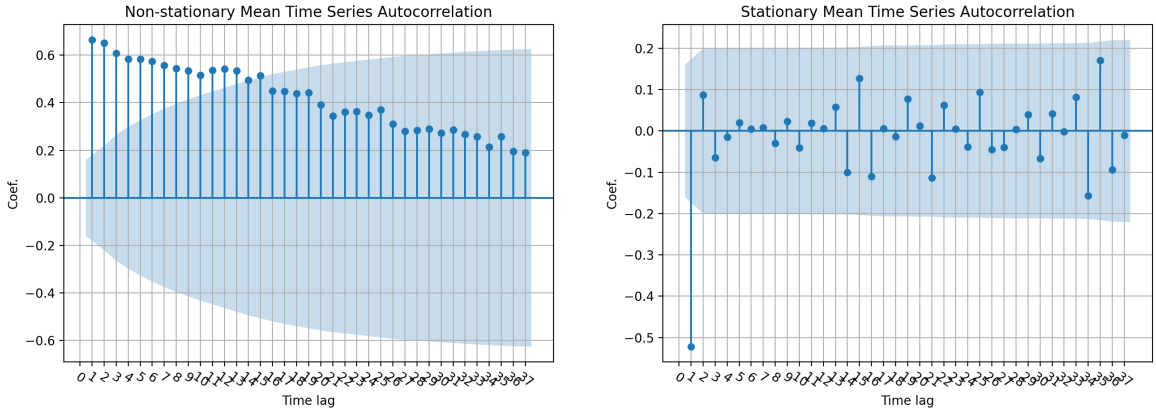
$$z_t = (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) - ... - (y_{t-d+1} - y_{t-d}) \tag{16}$$

In statistics, this order is also referred to as the order of integration (hence the "I" in ARIMA).

When trying to formulate a tentative ARIMA model for some time series y in a practical setting and decide whether differencing might be needed, a plot of the sample Autocorrelation Function (ACF) of y is often consulted. The ACF provides the autocorrelation coefficient $r_k(y)$ at time lag k and can be calculated as:

$$r_k(y) = \frac{\sum_{t=1}^{n-k}(y_t - \overline{y})(y_{t+k} - \overline{y})}{\sum_{t=1}^{n-k}(y_t - \overline{y})^2} \tag{17}$$

, where n is the length of y and it is assumed that y has not been mean centered.



(a) ACF (Original series)

(b) ACF (Differenced series)

Figure 9: Autocorrelation Function (ACF) for a synthetically created stochastic series with a deterministic linear trend. Figure (a) shows the ACF calculated from the original series. Figure (b) shows the ACF after applying first order differencing to the original series. The y-axis is the value of the autocorrelation coefficient at the time lag shown on the x-axis. The grey area shows the 95 % confidence interval.

If y has a non-stationary mean, the autocorrelation coefficients will somewhat slowly decay towards insignificant values as the time lag k increases. This is illustrated in Figure 9, which shows a synthetically created stochastic series with a deterministic linear trend before (a) and after (b) first order differencing has been applied. The slow decay of autocorrelation coefficients is evident in sub-figure (a), while sub-figure (b) show that no significant autocorrelation except for the coefficient at lag 1 remains. Note that such a negative spike with a value of ≥ 0.5 can indicate the presence of over-differencing, especially when the order of differencing is > 1. It may also call for the use of an MA(1) model, which

can exploit this remaining (now negative) autocorrelation in the differenced series z between z_t and z_{t-1}.

An ARIMA model combines the AR and MA models with differencing and is often denoted as ARIMA(p, d, q), where p denotes the order of the AR component, d the order of integration and q the order of the MA component. The general ARIMA(p, d, q) model has the form:

$$z_t = c + \epsilon_t + \sum_{i=1}^{p} \phi_i z_{t-i} - \sum_{i=1}^{q} \theta_i \epsilon_{t-i} \tag{18}$$

Where z_t is the d-order difference of the original time series y_t.

Seasonality in a time series refers to a periodic variation in the values. If this property occurs, it may be useful to consider it in an ARIMA model, e.g., to induce stationarity or include an auto-regressive or moving average term that matches the seasonality. If a time series has a seasonal component with s periods per season, a matching purely seasonal ARIMA model is denoted as ARIMA$(P, D, Q)_s$. Such a model has the form:

$$w_t = c + \epsilon_t + \sum_{i=1}^{P} \phi_i w_{t-is} - \sum_{i=1}^{Q} \theta_i \epsilon_{t-is} \tag{19}$$

Where w_t is the D-order seasonal difference of the original time series y_t. The D-order seasonal difference is computed similarly to the ordinary difference:

$$w_t = (y_t - y_{t-s}) - (y_{t-s} - y_{t-2s}) - ... - (y_{t-(D-1)s} - y_{t-Ds}) \tag{20}$$

The ACF plot for a time series with a suspected seasonal component can be consulted to aid in deciding whether seasonal differencing might be called for, as was the case with ordinary differencing. Now however, the factor indicating the need for differencing is a slow decay of the autocorrelation coefficient values at lags that are multiples of s.

If a time series has both seasonal and non-seasonal components, it may be best estimated by a mixed ARIMA$(p, d, q)(P, D, Q)_s$ model. Mixed ARIMA models are most conveniently described using the concise so-called backshift notation. For this purpose, we introduce the backshift operator B^i, which, when multiplied with any time-subscripted variable x_t, denotes lagging the variable i time steps:

$$B^i(x_t) = x_{t-i} \tag{21}$$

We can then express the AR(1) model as:

$$(1 - \phi B)y_t = c + \epsilon_t \tag{22}$$
$$y_t - \phi y_{t-1} = c + \epsilon_t \tag{23}$$

Similarly, the MA(1) model in backshift notation is:

$$y_t = c + (1 - \theta B)\epsilon_t \tag{24}$$
$$y_t = c + \epsilon_t - \theta \epsilon_{t-1} \tag{25}$$

Note that B behaves like an ordinary algebraic variable when multiplied by itself, meaning that $\forall i, j \in \mathbb{N}\, B^i B^j = B^{i+j}$.

For brevity, let us further denote the general seasonal and non-seasonal AR and MA operators in the following fashion:

$$\phi(B) = (1 + \phi_1 B + \phi_2 B^2 + ... + \phi_p B^p) \quad \text{(AR}(p)\text{)} \tag{26}$$
$$\theta(B) = (1 + \theta_1 B + \theta_2 B^2 + ... + \theta_p B^q) \quad \text{(MA}(q)\text{)} \tag{27}$$
$$\phi(B^s) = (1 + \phi_s B^s + \phi_{2s} B^{2s} + ... + \phi_{Ps} B^{Ps}) \quad \text{(Seasonal AR}(P)\text{)} \tag{28}$$
$$\theta(B^s) = (1 + \theta_s B^s + \theta_{2s} B^{2s} + ... + \theta_{Qs} B^{Qs}) \quad \text{(Seasonal MA}(Q)\text{)} \tag{29}$$

Lastly, let us introduce the differencing operator ∇:

$$\nabla^d = (1 - B)^d \quad (d\text{-order difference}) \tag{30}$$
$$\nabla_s^D = (1 - B^s)^D \quad (D\text{-order seasonal difference}) \tag{31}$$

We can now express the general mixed ARIMA$(p, d, q)(P, D, Q)_s$, which multiplicatively combines the seasonal and non-seasonal terms, in backshift notation:

$$\phi(B^s)\phi(B)\nabla_s^D \nabla^d y_t = c + \theta(B^s)\theta(B)\epsilon_t \tag{32}$$

To establish a baseline for ARIMA models, two seasonal ARIMA models were automatically generated using the `auto_arima` method [1] from the statistical *Python* library `pmdarima`, which is a *Python* implementation of a similar method found in the *R forecast* package [25]. This method automatically determines the appropriate ARIMA model order based on the Akaike Information Criterion (AIC). The AIC, when applied to ARIMA models, is defined as:

$$AIC = -2ln(L) + 2(p + q + P + Q + 1), \tag{33}$$

where L is the maximized likelihood of the model fitted to differenced time series and the number 1 denotes the optional constant c. The order of difference is determined by conducting multiple iterations of the KPSS unit-root test [28]. To avoid having to consider all possible combinations of p, d, q, P, D and Q, the model order is determined by the step-wise algorithm presented in [25], in which p and q are constrained within the interval $[0, 5]$, while P and Q are constrained within the interval $[0, 2]$. These intervals align well with the principle of parsimony for manually identifying the best suited model order, which mandates that the goal should be to find the model that contains the fewest possible coefficients to adequately explain the underlying process of the observed data. The resulting model orders for DK1 and DK2 were ARIMA(5, 0, 0)(0, 0, 2)$_{12}$ and ARIMA(1, 0, 0)(0, 1, 0)$_{12}$, respectively.

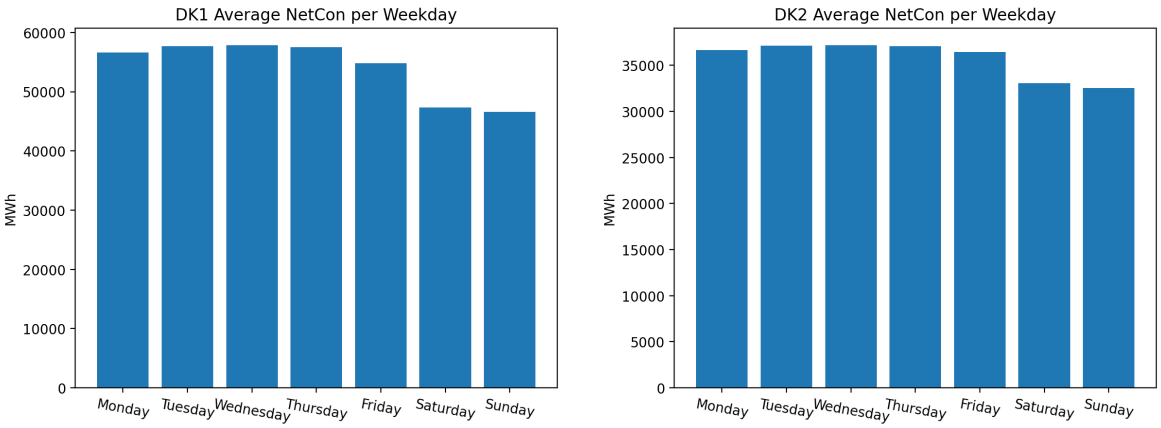
Following the application of the two automatically generated models, two manual ones were drafted, with the one originally fit to DK1 data being an ARIMA(0, 0, 1)(0, 1, 1)$_{12}$ and the one fit to DK2 data an ARIMA(0, 0, 0)(2, 1, 1)$_{12}$ model. They were created by inspecting the ACF plots (such as the ones in Figure 9) and the partial ACF plots, a process described in [33]. Through experimentation, it was later found out that the single MA terms in the DK1 model could favorably be replaced by multiple AR terms, reducing training error by 7 % and test error by 2.5 %, at the cost of parsimony. This more accurate model had the order ARIMA(4, 0, 0)(5, 1, 0)$_{12}$.

Interestingly, while both the time series plots themselves and their ACF indicated that their means were non-stationary and seasonal differencing might be required to induce stationarity, the automatically generated model for DK1 does not include any differencing at all.

6.1.3 ARIMA with Workday-Normalized Electricity Consumption

The electricity consumption data used in this project requires summation to accumulate data at a monthly time resolution. However, this introduces an irregularity to the accumulated data: Not only does the number of days per month vary, but so does the number of workdays per month. Hence, the accumulated data no longer occurs at equally spaced time intervals, and a given difference in electricity consumption in one month versus another may simply be due to a difference in workdays. Therefore,

a normalization approach may be called for to bring the data at all time steps to a common scale. In [33] it is suggested that one such normalization technique could be to simply divide each month by the number of workdays in said month, to bring each month to a per workday scale. This is based on the assumption that the composition of workdays is unimportant, i.e., the amount of each individual workday (Monday, Tuesday, .., Friday) does not matter or, said otherwise, electricity consumption is not affected by the type of workday. Analysis of the mean electricity consumption per weekday revealed that this was approximately true for both DK1 and DK2 (See Figure 10), although more so in DK2 than in DK1. After applying this normalization, an ARIMA model may be fitted to the resulting series and predictions generated by this model brought back to the original scale by multiplying them by the number of workdays in the corresponding month.



(a) DK1

(b) DK2

Figure 10: Bar plots showing the average net electricity consumption per weekday for the two price areas DK1 and DK2.

6.1.4 Pattern Sequence Based Forecasting

In 2008, Álvarez et al. presented a new algorithm [31] called Label-based Forecasting (LBF), which combines clustering and string matching to generate predictions for a given time series $\vec{s} \in \mathbb{R}^m$. It was later modified and renamed Pattern Sequence Based Forecasting (PSF) [12]. PSF first divides the time series into equal-sized segments $X = \{\vec{x_1}, \vec{x_2}, ..., \vec{x_n}\}, \vec{x_i} \in \mathbb{R}^c$, called cycles, whose length c should be equal to an existing periodic component of the series. In the original paper, which applied the PSF algorithm to a time series consisting of hourly observations, c was chosen to be 24, corresponding to one day. Afterwards, k-means clustering is applied to the sequence of vectors X to obtain a scalar sequence $L = \{l_1, l_2, ..., l_n\}$ of cluster labels. The hyperparameter k, determining the number of clusters, is chosen via a simple search within predefined upper and and lower bounds, where for each k, the k-means clustering algorithm is run and one or more cluster metrics are used to evaluate the resulting clustering. Afterwards, the k that resulted in the best cluster metric score is kept for all further clustering. While the first LBF algorithm used only the Silhouette index, the modified PSF computes a majority score based on three cluster evaluation metrics (Silhouette-, Dunn-, and Davies-Bouldin index). However, the paper on PSF [12] fails to showcase an improvement in prediction accuracy as measured by the mean relative error, as both the original LBF and PSF achieve the same error scores on the same datasets. Therefore, and to speed up computation, the *Python* implementation used in this thesis relies solely on the Silhouette index to find k.

To predict the next cycle $\vec{x}_{n+1}$ of the time series, the pattern that is made up by the previous W labels in L are considered, where W is called the window size. L is searched for all matching subsequences that exactly match this pattern, and the index of each single label following a match is retained. These indices are then used to retrieve the corresponding cycles in X, whereupon their average is computed to obtain $\vec{x}_{n+1}$. If more predictions are needed, the latest predicted cycle is appended to X and the algorithm is repeated. The following equations show how a prediction is computed:

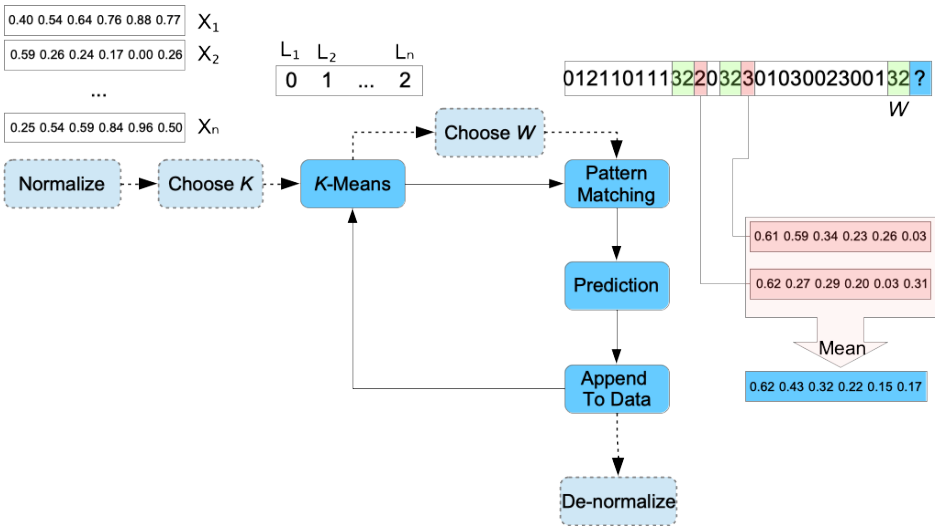


Figure 11: Diagram illustrating the PSF algorithm. Lighter shaded rectangles and dotted arrows represent operations that are only executed once, while the darker shaded rectangles and solid arrows represent the prediction loop, which is executed for each new prediction.

$$P = L_{n-W}^{n} \tag{34}$$

$$E_P = \{i \,|\, L_{i-(W+1)}^{i-1} = P,\, i < n\} \tag{35}$$

$$\vec{x}_{n+1} = \frac{1}{|E_P|} \sum_{i \in E_P} X_i \tag{36}$$

Similar to finding k, W is also determined by searching a predefined interval of values. To this end, X is split in time into a training and a validation set, where the training set makes up 70 % of X. Thereafter, for each candidate W, a number of cycles equal to the size of the validation set are predicted and the mean absolute error is used to evaluate the prediction performance. The W resulting in the lowest prediction error is then kept for all further forecasts. In [12], 12-fold cross validation was used to determined W, with each fold corresponding to one month of data, whereas [31] used a form of leave-one-out cross validation, where the total absolute error between the 1 step ahead prediction of each cycle in X and its observed value was minimized. If no matching pattern of size W can be found, W is decremented and the search repeated. Figure 11 shows a diagram illustrating the PSF algorithm and highlights the prediction loop.

As evident in Figure 11, the data is normalized before applying PSF. In [12], each hour in a cycle is divided by the cycle mean, but reverting this operation would require knowing the mean of every future predicted cycle, which is not feasible in a real world prediction scenario. Instead, simple min-max normalization is applied:

$$\vec{s}_{norm} = \frac{\vec{s} - min(\vec{s})}{max(\vec{s}) - min(\vec{s})} \tag{37}$$

Where $\vec{s}_{norm}$ is the normalized series. If the minimum of $\vec{s}$ is negative, its absolute value is added to all values of $\vec{s}$ before normalizing. The min-max normalization approach was also taken by Bokde et al. [14], who wrote a software library, written in the R programming language, that implements a variant of PSF. The *Python* implementation used in this project is based on this library.

When predicting a cycle, it is possible that W reaches 0, e.g., when the previous cycle is an outlier and thus constitutes a cluster of size 1. This issue is discussed in neither [31] nor [12], but [14] introduces a fallback option that aims to solve this problem by simply using the last (outlying) cycle as the prediction. However, this method can have the undesirable effect that the same cycle is used for all predictions. Another approach could be to use the average of all preceding cycles as the prediction.

While this would solve the aforementioned repetition issue, this prediction could also be sub-optimal, e.g., if the time series contains a few extraordinary cycles whose values are significantly larger or smaller than the average cycle. As these cycles are unusual, it might be desirable to discard them before computing a mean to use as the prediction. This can be achieved by using the centroid of the largest cluster as the predicted next cycle, an approach presented in [30], where Majidpour et al. present a modified PSF algorithm that combines PSF and elements from the k-nearest neighbors algorithm. A comparison of three separate PSF algorithms that only differed in the choice of fallback method, conducted as part of this thesis, showed that for the Danish weather parameter and electricity consumption time series, the algorithm employing the centroid based fallback method achieved the lowest mean absolute prediction error for electricity demand and in the large majority of weather parameter cases. Hence, it was picked as the fallback method for the *Python* implementation of PSF.

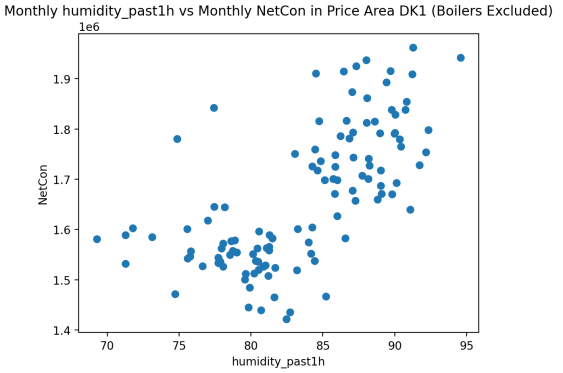
While both the original articles on LBF and PSF and multiple others mention a cycle length $c = 24$ for hourly electricity demand time series, trying to follow their example for the times series at hand yielded unusable predictions, which did not even follow the basic annual seasonal pattern and achieved a vastly lower forecast MAE than the naive models described in section 6.1.1. This result was shared between the R implementation of PSF and the custom *Python* implementation written for this project. Inspecting the label series L revealed that the k in k-means was chosen to be only 2, leading to a binary labeling of the cycles. The lack of discrimination between cycles resulted in a feedback loop, where after a short amount of time, all matching patterns for future predictions were found in the section of L corresponding to previously predicted cycles. This manifested itself as step-like patterns in the prediction plots that would eventually converge to a straight horizontal line. In [30], Majidpour et al. encountered a similar problem and attempted to mitigate it by setting the lower bound of k to 10 % of distinct days in the data and the maximum k equal to the number of distinct days. However, it is unclear as to how this lower bound was chosen. In addition, for the data used in this thesis, this approach is practically infeasible when run on a laptop computer, as 2547 possible values of k reaching from 282 to 2829 would need to be considered, resulting in a running time of multiple hours for a single time series. Instead, a modification that finally led to sensible predictions was to set the cycle length c equal to 1 year, or 8760 hours when working with an hourly temporal resolution. Although, logically, this solution should reduce the algorithm's ability to capture finer details in the data, e.g., by clustering holidays and other high-consumption days together, it still yielded competitive prediction accuracy when compared to the other approaches investigated in this project. Later, it was discovered that the most likely reason for the PSF algorithm failing initially to produce usable predictions when working on hourly data with $c = 24$ was the annual seasonality of the data. Simply computing the first order seasonal difference of the hourly electricity consumption data before applying PSF alleviated the issue.

6.2 Multivariate Models

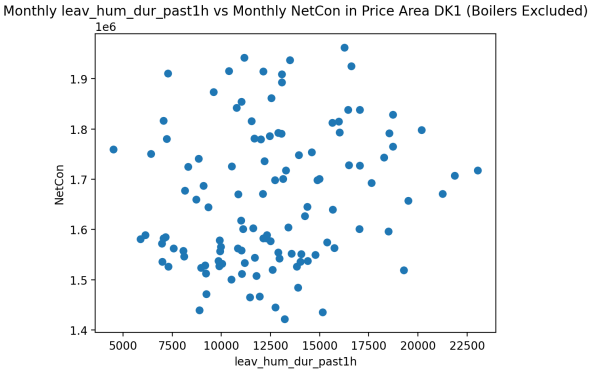
6.2.1 Motivation for Linear Models

Before applying linear models to predict electricity consumption based on weather parameters, it seems appropriate to explain the motivation for applying them in the first place. In particular, a necessary condition for these models to produce good results is that the relationship between the independent variables and the dependent variable is at least somewhat linear. Historically, this relationship has been found to be sufficient in select regions, as several research papers show that accurate results have been achieved using linear models [32] in the domain of electricity consumption prediction, with some finding that they can even achieve lower prediction errors than artificial neural networks [23].

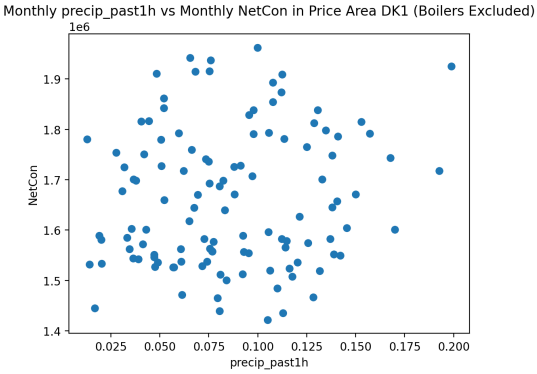
The choice of model should always be driven directly by the available data. In other geographic regions, such as the USA, both the climate itself and the landscape of electricity consuming devices differs from what can be found in Denmark. The existence of subtropical weather in parts of the USA combined with the prevalence of residential air-conditioning has a significant effect on the relationship between select weather parameters and electricity consumption. While plots illustrating the relationship between temperature and demand in Denmark show what appears to be a somewhat linear relationship that levels out towards the highest temperatures, some plots using data from the USA show a more parabola-like shape, in which the demand begins to rise again after a certain temperature is reached.



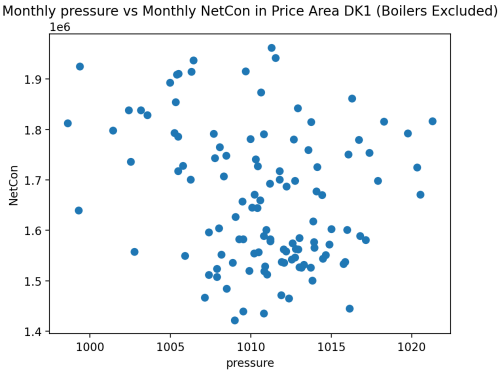
(a) Humidity



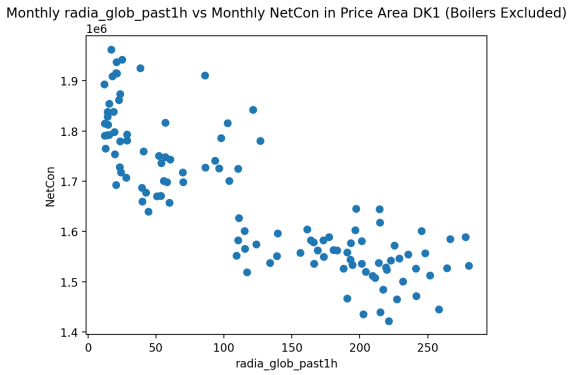
(b) Leaf Humidity Duration



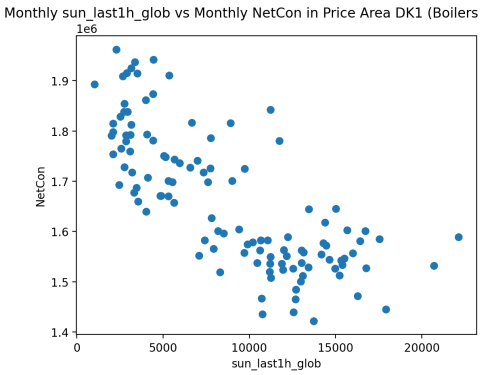
(c) Precipitation



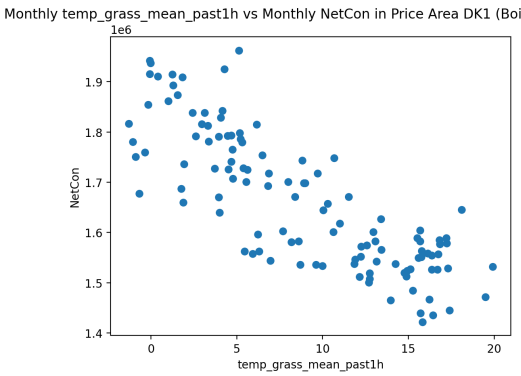
(d) Pressure



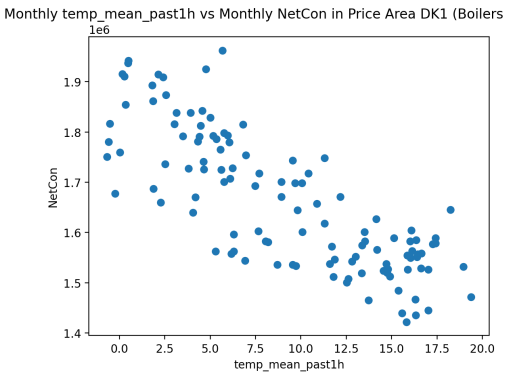
(e) Global Irradiance



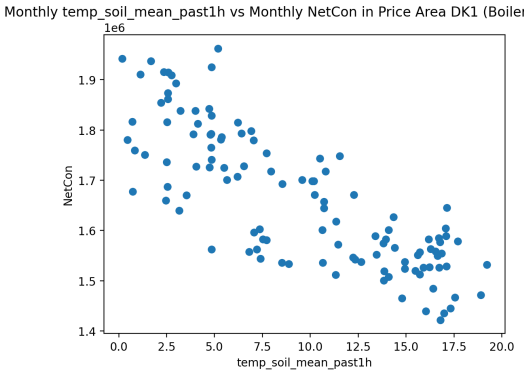
(f) Sunshine Duration



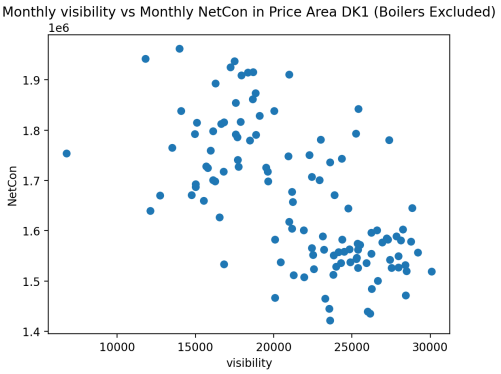
(g) Grass Temperature



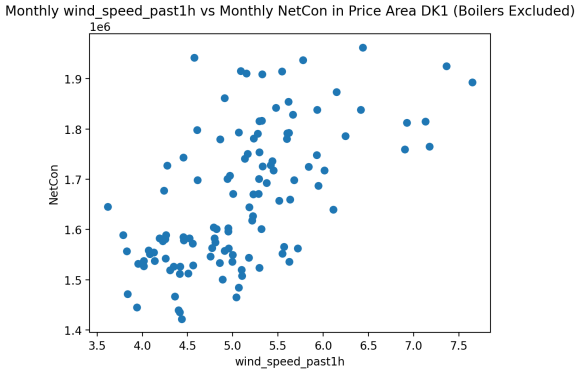
(h) Temperature



(i) Soil Temperature



(j) Visibility



(k) Wind Speed

Figure 11: Scatter plots of each weather parameter, downsampled to a monthly time resolution, versus the total monthly electricity consumption of the price area DK1.

Figure 11 shows the relationship between each of the 11 weather parameter used in this project and electricity consumption. The data is from price area DK1 and has been downsampled to a monthly time resolution, since all multivariate models use monthly input data. While little correlation is visible in the plots of leaf humidity, precipitation and pressure, it appears that the point clouds of the other parameters could be fit fairly well by a straight line. A similar plot for DK2 data can be found in the appendix, together with plots showing the relationships at daily and hourly time resolutions. In the following subsections, multiple linear models, which were applied to determine whether this early observation holds any truth, will be described.

6.2.2 Linear Regression

One of the first modelling approaches for energy consumption forecasting was linear regression [19]. It enables the estimation of a target variable as a linear function of a number of independent random variables. In 1958, for example, an attempt was made to predict future energy consumption based on 5 weather variables [16]. Since the relationship between weather parameters and electricity consumption can be non-linear [32], depending on the geographic region and temporal resolution of the data, it might be necessary to include quadratic or even cubic polynomials of the independent variables in the model. Even though the forecasting performance of multiple linear regression models can improve upon simple ones, they may be limited by their simplicity when compared to, e.g., Artificial Neural Networks (ANNs), which can extract arbitrary functional relationships [34]. In addition, they require careful and time consuming feature engineering to reach good results. Choosing the optimal features is not limited to selecting suitable input variables by themselves, but also determining dynamic temporal relationships. For example, a model using temperature as an input variable may need to include several time-lagged versions of it to best approximate the relationship between several consecutive days of low temperatures and a subsequent lagged increase in energy consumption.

An important benefit of simple and multiple regression models is that they assign easily comparable

coefficients to their input variables. These coefficients make the models highly interpretable by humans, since they directly explain the influence that each feature has on the target variable. For univariate regression models, this influence can also be inferred from training the model on a training data set and then calculating one or more performance metrics on model predictions against a test data set. The sign and magnitude of the coefficient in a univariate regression model can also be used to establish the correlation between the independent and the dependent variable. Even if the independent variables have been transformed, for example through the use of a dimensionality reduction method or temporal downsampling, the person developing the regression model still retains a clear understanding of how the model arrives at its predictions. This is mostly not the case when black box models such as ANNs are used to automatically extract features and subsequently establish the relationship between them and the target variable.

6.2.3 Independent Variable Correlation Analysis

Univariate linear regression was used to analyze the linear correlation between the different weather parameters and electricity consumption in the two price areas. This method was chosen over basic correlation analysis, since it is more in line with what we wish to establish, namely which weather parameters can be used to forecast electricity consumption. The slope coefficient of the independent variable in each regression model was used to determine whether the corresponding weather parameter is positively, negatively or not at all correlated with electricity consumption, denoted by a positive, negative or zero coefficient, in that order. For the analysis, the weather parameters were averaged geographically for the two price areas and temporally to reach a monthly time resolution. To enable the direct comparison of coefficients from models based on different weather parameters, the inputs of each model were normalized by subtracting the mean and dividing by the L_2-norm, i.e.:

$$\vec{z} = \frac{\vec{x} - \mu}{||\vec{x}||_2} \tag{38}$$

Where $\vec{z}$ is the normalized input vector, $\vec{x}$ is the original input vector and μ is the mean of $\vec{x}$. After normalizing, the univariate regression models used in this analysis had the form:

$$\hat{y}_t = \beta_0 + \beta_1 z_t + \epsilon_t \tag{39}$$

Where β_0 and β_1 are the intercept and slope coefficient, respectively, whose values were estimated using ordinary least squares (OLS), $\hat{y}_t$ is the dependent variable at time step (in this case, month) t, i.e., the predicted energy consumption and ϵ_t is the error term.

The correlations, indicated by the normalized coefficients, were found to be in agreement across the two price areas and relatively close in value, taking into account the difference in energy consumption between DK1 and DK2. Since the coefficients were estimated based on normalized data, their values can be interpreted as the effect that a one unit increase in the normalized independent variable has on the dependent variable. The parameters humidity, leaf humidity duration, precipitation, snow depth and wind speed were all positively correlated with electricity consumption, while pressure, solar radiation, sunshine duration, air/grass/soil temperature and visibility all showed a negative correlation with electricity consumption. However, the slope coefficients in the cases of precipitation (68 386.22 and 35 046.72), leaf humidity duration (128 762.98 and 131 950.89) and pressure ($-177\,636.79$ and $-116\,692.3$) were small compared to all other variables, whose coefficients had absolute values in the range [403 501.08, 1 056 849.14] (See Table 3).

In addition to the coefficients themselves, t and p values were calculated for each to test their statistical significance. The null hypothesis was $H_0 : \beta_1 = 0$, i.e., there is no linear correlation between the independent and the dependent variable. This analysis revealed that the coefficients in the case of precipitation were not significant at a significance level of $\alpha = 0.05$, meaning that the null hypothesis could not be rejected. However, computing the residual autocorrelation function revealed that there was a significant amount of autocorrelation present in the residual series. Since the statistical analysis is in part based on the assumption that residuals are mutually independent, the t and p values are invalid and cannot be used to determine whether or not the corresponding coefficients are significant.

Table 3: Slope coefficients resulting from using each weather parameter as the independent variable in a simple regression model, with electricity consumption being the dependent variable. Shown separately for the price areas DK1 and DK2. Independent variables were normalized to allow direct comparison between parameters.

Parameter	Coefficient (DK1)	Coefficient (DK2)
humidity_past1h	871608.38	820394.36
leav_hum_dur_past1h	128762.98	131950.89
precip_past1h	68386.22	35046.72
pressure	−177636.79	−116692.3
radia_glob_past1h	−1054379.68	−955420.81
snow_depth_man	441165.61	403501.08
sun_last1h_glob	−961451.72	−910993.25
temp_grass_mean_past1h	−1056849.14	−1004071.45
temp_mean_past1h	−1034229.88	−985991.13
temp_soil_mean_past1h	−1037025.67	−984534.37
visibility	−800864.80	−783699.11
wind_speed_past1h	740823.82	775642.62

6.2.4 Linear Regression Using Single Weather Parameters

Each of the linear regression models initially used only a single weather parameter with no time lag as the independent variable. It was found, though, that in the case of DK1, including time as a second independent variable in the majority of cases (7/11) improved the forecasting error as measured by the Mean Absolute Error (MAE), which was calculated on the predictions vs the actual values of the test data set. The parameters for which the MAE did not improve are pressure, global irradiance, sunshine duration and wind speed. For DK2, the results when including time were similar: The MAE improved roughly half of the cases (6/11), namely parameters humidity, precipitation, pressure, solar radiation, air temperature and wind speed, but worsened in the cases of leaf humidity duration, sunshine duration, grass/soil temperature and visibility. Both the independent variables and the dependent variable were sampled at a monthly time resolution for these tests.

Single weather parameter based predictions were calculated both with a time lag of 0 and a time lag of 1 month. It was found that predictions based on solar radiation generally resulted in the smallest MAE on the test data set. Interestingly, the minimum MAE for DK1 was achieved using a time lag of 1, while the minimum for DK2 used no time lag on the independent weather variable. Thus, the following multiple OLS regression model was used to generate the most accurate energy consumption predictions based on a single weather variable for DK1:

$$\hat{y}_t = \beta_0 + \beta_1 z_{t-1} + \beta_2 x_t + \epsilon_t \tag{40}$$

Where $\hat{y}_t$ is the predicted value of the independent variable at time step t, z is the normalized independent weather variable, x is the independent time variable (given as a floating point number) and ϵ the error term.

Similarly, the model yielding the most accurate forecasts for DK2 using solar radiation as the independent variable was:

$$\hat{y}_t = \beta_0 + \beta_1 z_t + \beta_2 x_t + \epsilon_t \tag{41}$$

Note that z is not lagged in this model.

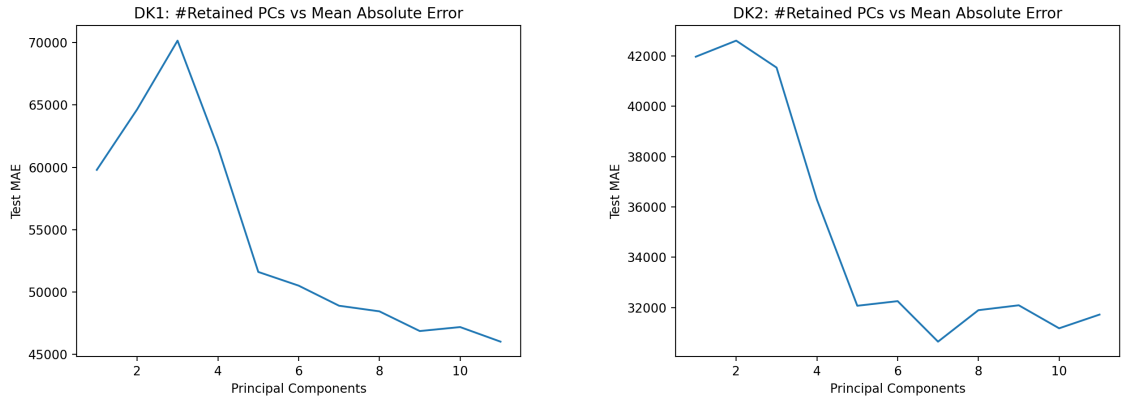
6.2.5 Principal Component Regression Using Multiple Weather Parameters

Moving from simple linear regression to multiple linear regression models, a choice had to be made on which weather parameters to include as regressors. While the correlation analysis had shown that precipitation, leaf humidity duration and pressure were much less correlated with electricity consumption than the rest of the variables, it could not be shown that they had no correlation whatsoever. In

addition, multiple parameters are highly linearly correlated, in particular the three temperature variables and the two solar-related parameters. This goes directly against the assumption of *independent* variables in a linear regression model. It was therefore not simply a case of variable selection, but instead a problem of finding a combination of all parameters, such that the multicollinearity between the resulting regressors be minimal. A common method for addressing this issue is Principal Component Analysis (PCA), which reduces the amount of variables by replacing them with a number of orthogonal principal components, each of which is a linear combination of the original variables.

Singular Value Decomposition (SVD) was used to perform PCA on a $n \times m$ matrix M, where n is equal to the number of time steps, i.e., the number of observations, and m equals the number of weather parameters. Since the snow depth time series is missing many observations, it was excluded from M. Following the PCA, a change of basis was applied to M using one or more principal components, yielding a new $n \times k$ matrix N, with $k \leq m$. Each column of N was then used as an independent variable in a multiple linear regression model together with time as a separate independent variable. The principal directions, or basis vectors associated with the principal components, as computed on weather data from DK1, are shown in Table 1 and continued in Table 2, both of which can be found in the appendix. A similar table for data from DK2 can also be found in the appendix. Since each dimension of a given principal direction vector corresponds to one of m weather parameters, we can analyze the magnitude of each dimension to determine the "importance" of the corresponding parameter for the principal component. For example, principal component 1, which explains 70.5 % of the total variance in the weather data, uses mainly data from sunshine duration, leaf humidity and visibility parameters to do so.

The change of basis and subsequent regression was performed with the first, the first and second, the first, second and third, et cetera, until all principal components were used. Thereafter, the number of principal components yielding the smallest MAE when calculated on the test data set was determined. Figure 12 illustrates the relationship between the number of included principal components and the MAE achieved by using them as independent variables in a multiple regression model. It is evident that for DK1 data, removing any principal component results in an increase in prediction error, meaning that the application of PCA in this case did not have the desired effect of reducing the number of regressors to achieve better forecasts. This effect did occur in case of the DK2 data, where the minimum in prediction error could be found at 7 retained principal components.



(a) DK1 (b) DK2

Figure 12: Number of principal components included in a linear regression model versus the resulting mean absolute error as computed on the test data set for the two electricity zone DK1 and DK2. Note that the y-axes of figures (a) and (b) differ.

6.2.6 Combined Prediction Models

While ARIMA, PSF and principal component regression models have been shown to be viable means of time series forecasting by themselves, this section will describe two multi-variable models that aim to

achieve a higher prediction accuracy than each of these stand-alone approaches, by combining ARIMA, PSF and dimensionality reduction based regression.

The general approach is shared between the two models: a PSF model is trained on the time series of each weather parameter to forecast the weather for a time period corresponding to the test data set. Thereafter, the forecasts are appended to the training data of each weather parameter and all time series are concatenated to form an $n \times k$ matrix M, where n is the total number of hours of weather observations and k the number of weather parameters. Next, dimensionality reduction is applied to M and the resulting $n \times d$ matrix M' is used as exogenous input to the `auto_arima` method (which is discussed in section 6.1.2).

Although the original intention was to obtain a Seasonal Auto-Regressive Integrated Moving Average model with eXogenous variables (SARIMAX) and use the model order found by the `auto_arima` method as a starting point, it became apparent that neither auto-regressive nor moving average coefficients seemed to be necessary to achieve the minimum AIC. All the most accurate models, found by varying spatial and temporal average computations, solely applied simple differencing, thus being equivalent to multiple linear regression using integrated variables. In addition, inspection of the residual autocorrelation plots revealed that no significant autocorrelation remained, apart from two spurious spikes at time lags 9 and 23 in the case of DK1 and 3 and 23 in the case of DK2.

The two models differed in the dimensionality reduction step, the first one applying Principal Component Analyis (PCA) and the second Principal Least Squares (PLS) regression. The use of PCA to regularize regressors is wide-spread in literature and is known as Principal Component Regression (PCR). The benefits of PCA for this purpose include implicit variable selection by minimizing the influence of variables with low variance, as well as combining highly correlated variables. Both of these are a direct result of mapping the original data to lower-dimensional space made up by a subset of principal components.

In other application areas such as clustering, the decision of how many principal components to retain is frequently based on the scree plot, in which the principal components are plotted against their eigenvalues, with the latter being in descending order. However, due to the objective of this project being prediction, a different kind of plot was chosen to determine the number of principal component to retain. Instead of the x-axis values representing individual principal components, they represent the number of retained principal components. Meanwhile, the y-axis shows the test MAE, obtained by using all retained principal components as regressors to make a forecast of equal length to the test set. This kind of plot can be seen in Figure 12. While the scree plot will ideally show a pronounced "elbow" in the line at a point where the gradient flattens out, the principal components vs. test MAE plot often times contains both an elbow, which may be more or less pronounced, and a minimum, after which including more principal components leads to overfitting or the inclusion of noise and thus an increase in test error. Hence, one may have to choose between using the elbow or the minimum as the decisive factor for choosing the number of principal components to retain. This can be seen as a trade-off between choosing the most parsimonious model (the elbow) and the most accurate one.

Despite the success of PCA as an universal dimensionality reduction method, it has an obvious limitation when used in conjunction with a regression model: It is unsupervised, i.e. it does not take into account the correlation between the independent variables and the dependent one. While one variable may have a lower variance than another, the former may have more predictive power w.r.t. the dependent variable. Thus, implicitly removing it would be a mistake. PLS regression is another method than can be used for dimensionality reduction. However, it is supervised in that it takes into account the cross-covariance between the independent variables and the dependent variable.

The variant of PLS regression used in this project applies SVD to the cross-covariance matrix of the input matrix X and the output vector y. Thereafter, the first left-singular and right-singular vectors are used as the input and output weights, respectively. It can be shown that the values that make up the first left-singular vector are proportional to the β-coefficients one would obtain via OLS-regression of y onto X [26]. The matrix X is then regressed on the input weights to obtain a vector of so-called input scores. Subsequently, X is "adjusted" for the first principal component, which is necessary to find subsequent components. This adjustment is achieved by subtracting the obtained input scores from the input matrix, a process known as "deflation", to obtain the residual matrix. The input matrix is effectively collapsed or deflated according to how much each variable contributes to the found principal

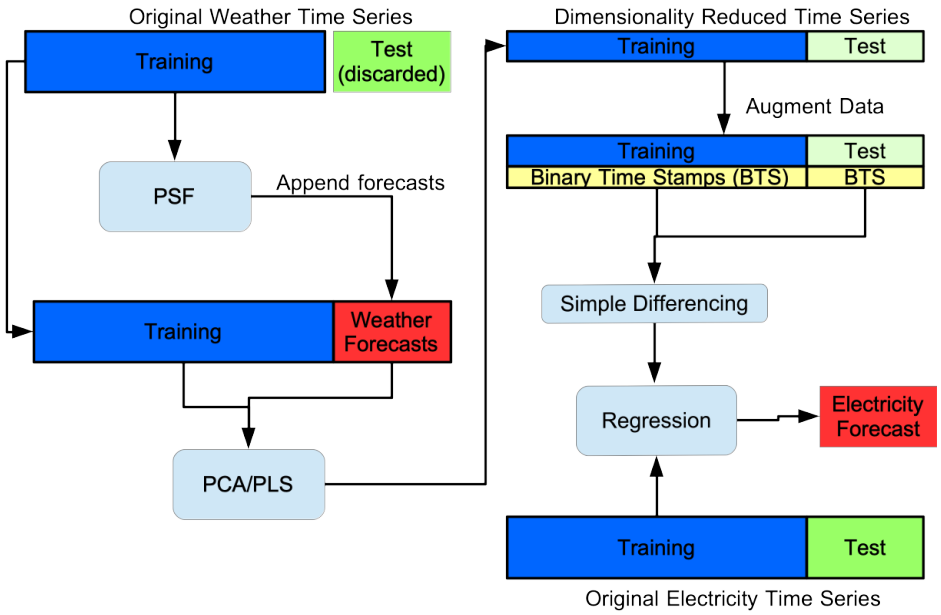


Figure 13: Diagram illustrating the steps taken in both of the combined prediction models to produce a forecast.

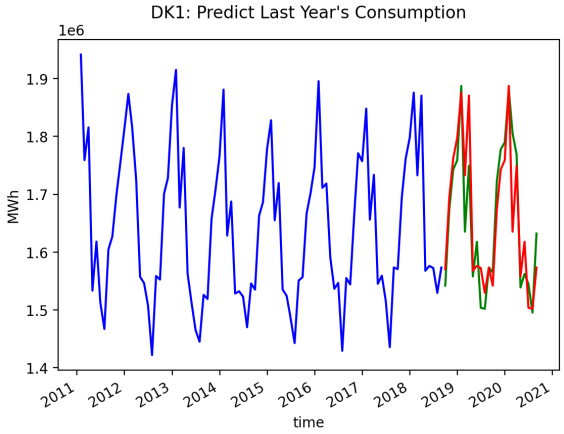
component, leaving behind data that is not explained by said component. Similarly, the output vector y is regressed on the input scores and it too is deflated using the output scores obtained through this process. The deflation marks the end of one iteration, meaning the process above is repeated for every subsequent principal component, with the new residual input matrix and residual output vector replacing the previous (residual) input data for the next iteration.

After either PCA or PLS is applied to transform the input matrix, additional independent indicator variables are appended to it. Their purpose is to model time and hence, depending on the temporal resolution, their number varies. At a monthly time resolution, only 11 indicator variables are used, with the 12th month being implicitly modeled via the intercept β_0 when all indicator variables are set to zero. When a daily time resolution is used, additional indicator variables modeling the weekdays are included. In case of an hourly time resolution, a further 23 indicator variables are added, representing the hours of each day. This method of modeling time allows the regression model to establish the general influence of each month (and optionally day/hour) on electricity consumption in conjunction with the effect of weather parameters for a more accurate prediction. The two combined models are illustrated by the diagram shown in Figure 13.

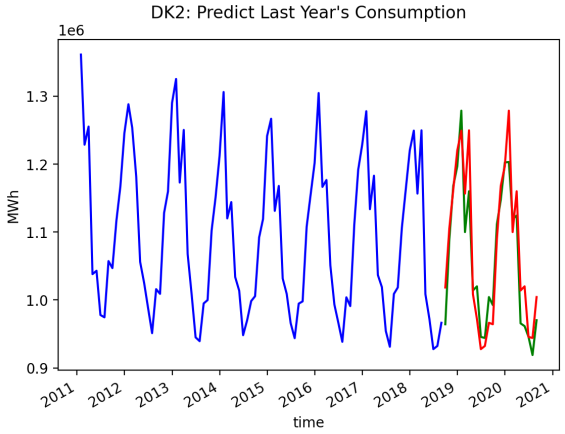
7 Prediction Results

7.1 Naive Baseline Models

For DK1, the MAE and MAPE (calculated on predicted versus actual values of the test data set) of the model repeating the consumption of the previous year were 45 334.24 MWh and 2.59 %, respectively, while they were 32 218.24 MWh and 3.03 % for DK2. Figure 14 shows time series plots of the actual representing energy consumption, split into a training and a testing set, and its predicted value. They can be used to evaluate how the models performed in the different parts of the test data set. The bar plots of Figure 15 are included to provide more specific insight into how each model performed on average in each month of the year, which can be a little difficult to pinpoint in the forecast plots. Note that the bar plots are mainly meant to facilitate intra-, not inter-model comparison of the MAE for each month. Thus, the y-axes of the bar plots can differ.

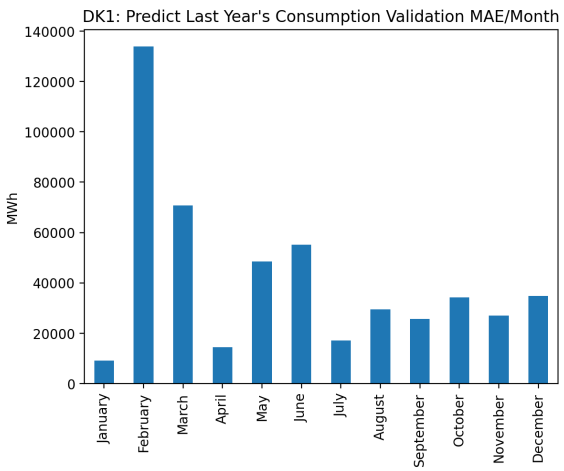


(a) DK1: Test MAE = 45 334.24 MWh, Test MAPE = 2.59 %

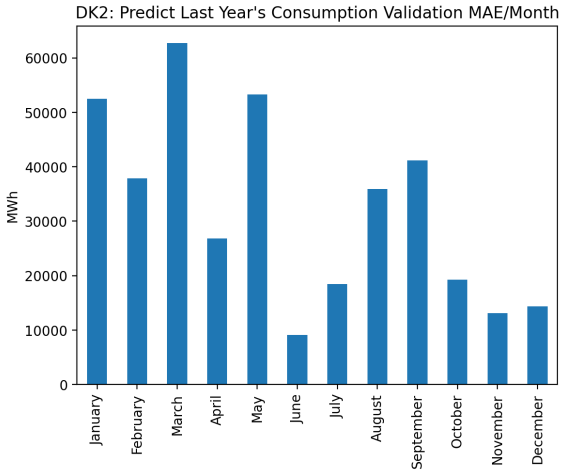


(b) DK2: Test MAE = 32 218.24 MWh, Test MAPE = 3.03 %

Figure 14: Electricity consumption training (blue) and testing (green) data with predicted consumption (red) for the two areas DK1 and DK2. Naive model repeating the previous year's consumption.



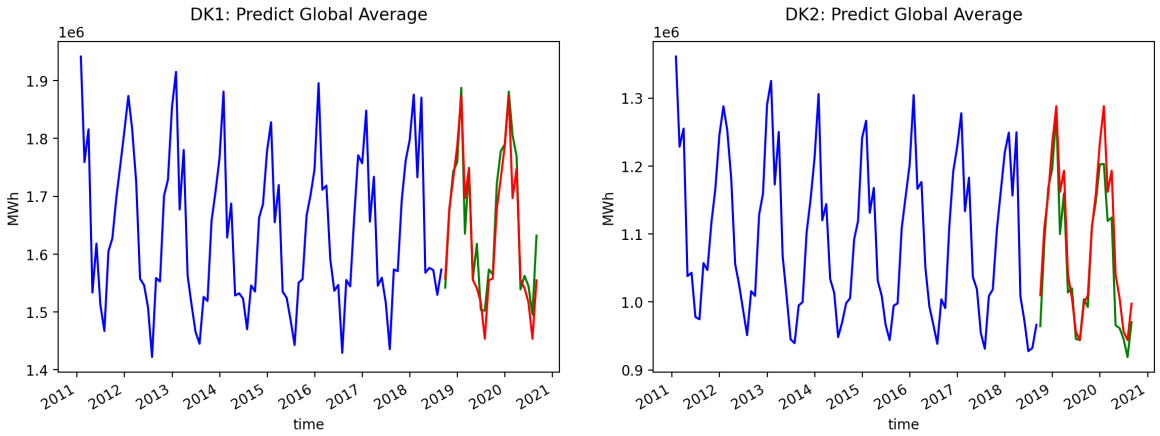
(a) DK1



(b) DK2

Figure 15: Average MAE per month calculated on validation/test set forecasts. Naive model repeating the previous year's consumption.

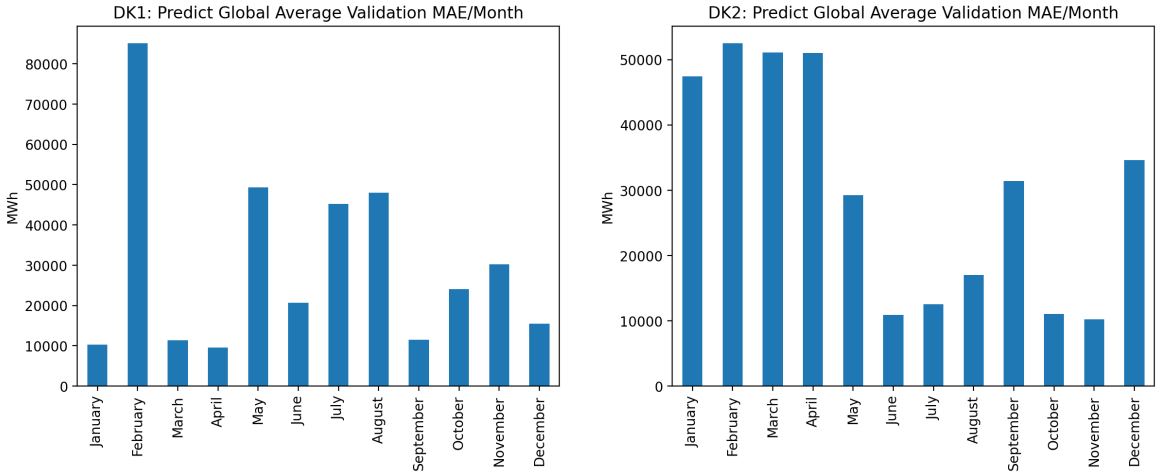
The second model, predicting the average of all years, achieved a test MAE of 49 950.94 MWh for DK1, while the test MAPE came out to be 2.9 %. For DK2, the test MAE was 27 901.35 MWh or 2.61 % as calculated by the MAPE.



(a) DK1: Test MAE = 49 950.94 MWh, Test MAPE = 2.9 %

(b) DK2: Test MAE = 27 901.35 MWh, Test MAPE = 2.61 %

Figure 16: Electricity consumption training (blue) and testing (green) data with predicted consumption (red) for the two areas DK1 and DK2. Naive model predicting the average over all years in the training set.



(a) DK1

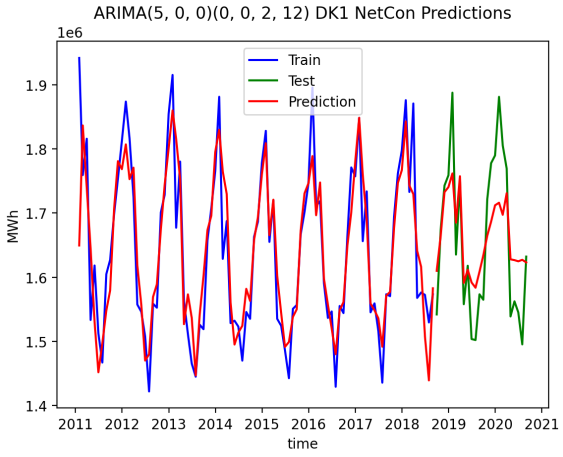
(b) DK2

Figure 17: Average MAE per month calculated on validation/test set forecasts. Naive model predicting the average over all years in the training set.

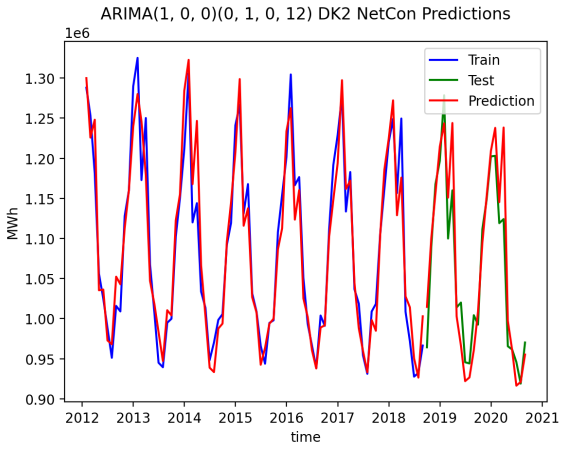
7.2 ARIMA

7.2.1 Auto-ARIMA

The automatically identified ARIMA$(5, 0, 0)(0, 0, 2)_{12}$ model fit to electricity consumption data from DK1 yielded a test MAE of 63 126.05 MWh, with the ARIMA$(1, 0, 0)(0, 1, 0)_{12}$ model fit to DK2 data achieving a test MAE of 29 464.94 MWh. Figure 18 showcases the forecasts generated by the two models.

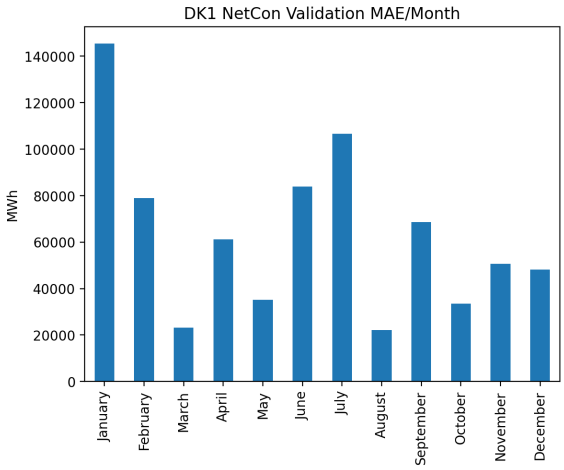


(a) DK1: Test MAE = 63 126.05 MWh, Test MAPE = 3.81 %

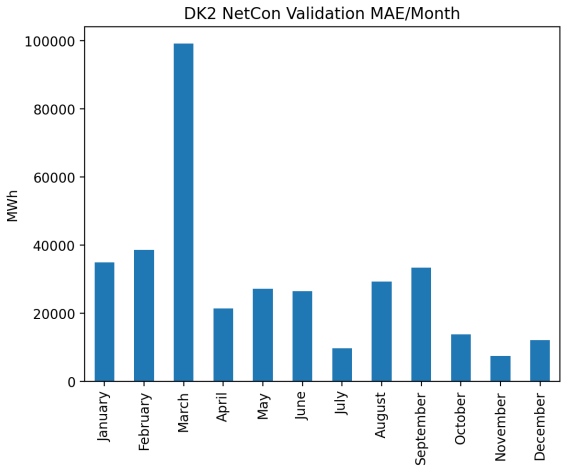


(b) DK2: Test MAE = 29 464.94 MWh, Test MAPE = 2.75 %

Figure 18: Electricity consumption training (blue) and testing (green) data with predicted consumption (red) for the two areas DK1 and DK2. Predictions calculated by automatically generated seasonal ARIMA models.



(a) DK1

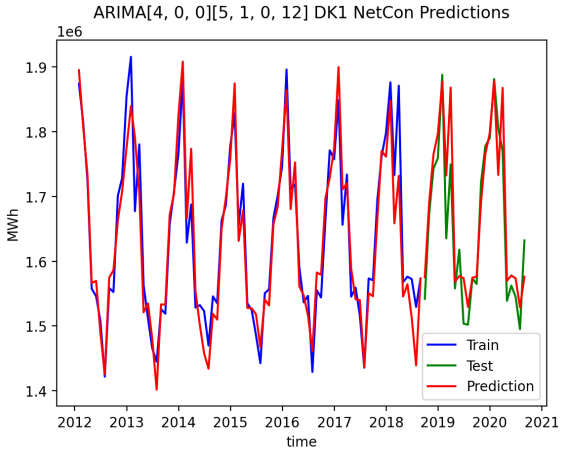


(b) DK2

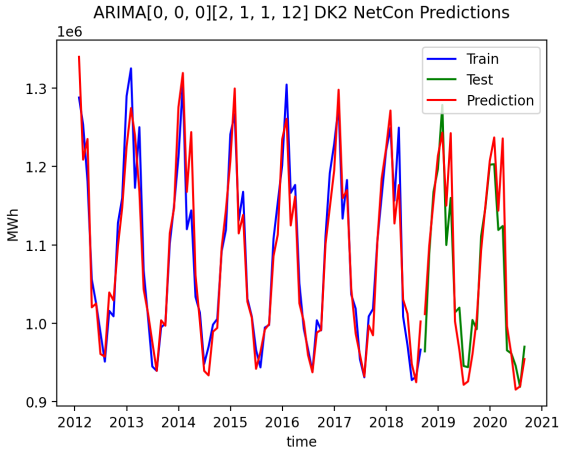
Figure 19: Average MAE per month calculated on validation/test set forecasts. Predictions calculated by automatically generated seasonal ARIMA models.

7.2.2 Manual ARIMA Models

The manually created models outperformed the automatically generated ones based on the MAE, with the DK1 model achieving a test MAE of 36 918.42 MWh and a MAPE of 2.23 % and the DK2 model obtaining a test MAE of 28 930.01 MWH and a MAPE of 2.7 %. The resulting forecast plots are illustrated in Figure 20.

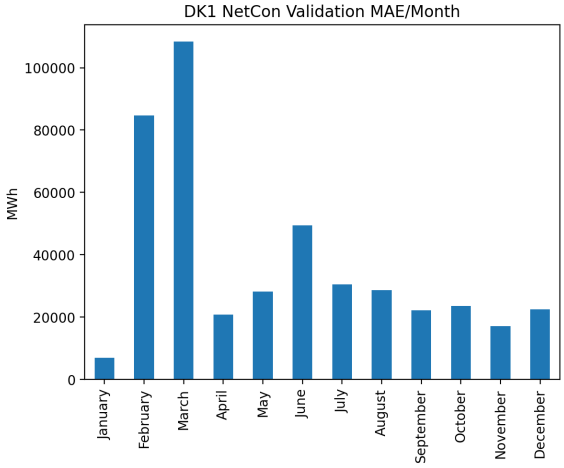


(a) DK1: Test MAE = 36 918.42 MWh, Test MAPE = 2.23 %

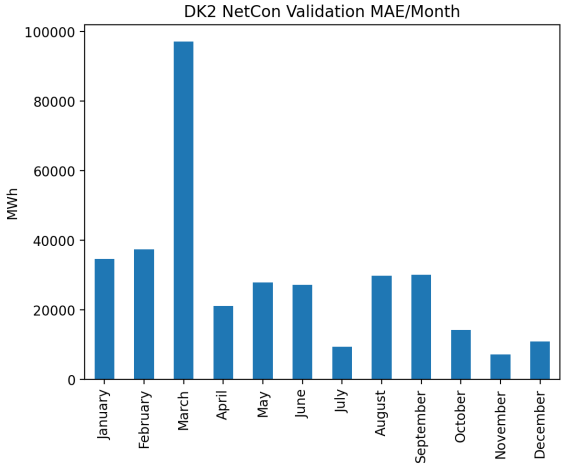


(b) DK2: Test MAE = 28 930.01 MWh, Test MAPE = 2.7 %

Figure 20: Electricity consumption training (blue) and testing (green) data with predicted consumption (red) for the two areas DK1 and DK2. Predictions generated by manually developed seasonal ARIMA models.



(a) DK1

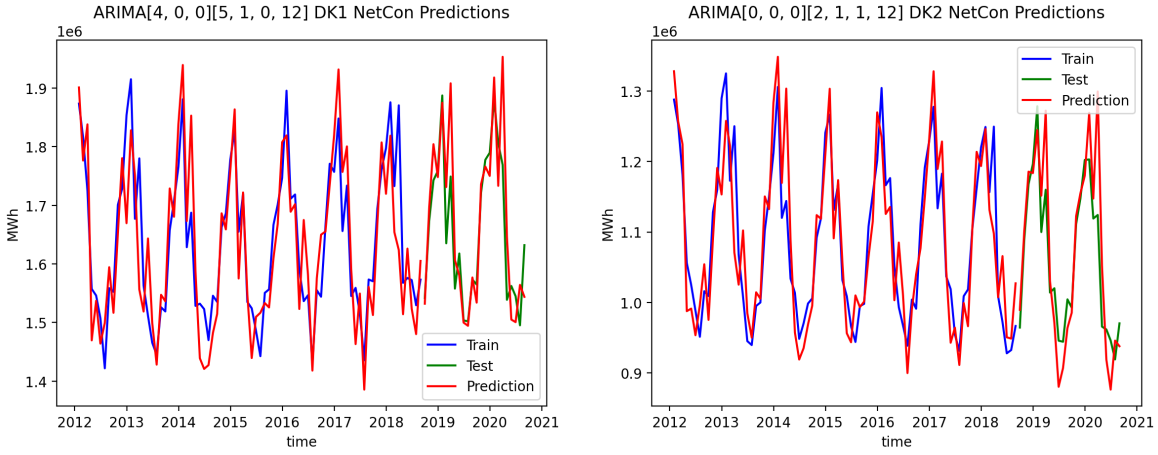


(b) DK2

Figure 21: Average MAE per month calculated on validation/test set forecasts. Predictions generated by manually developed seasonal ARIMA models.

7.2.3 ARIMA (Workday Normalized)

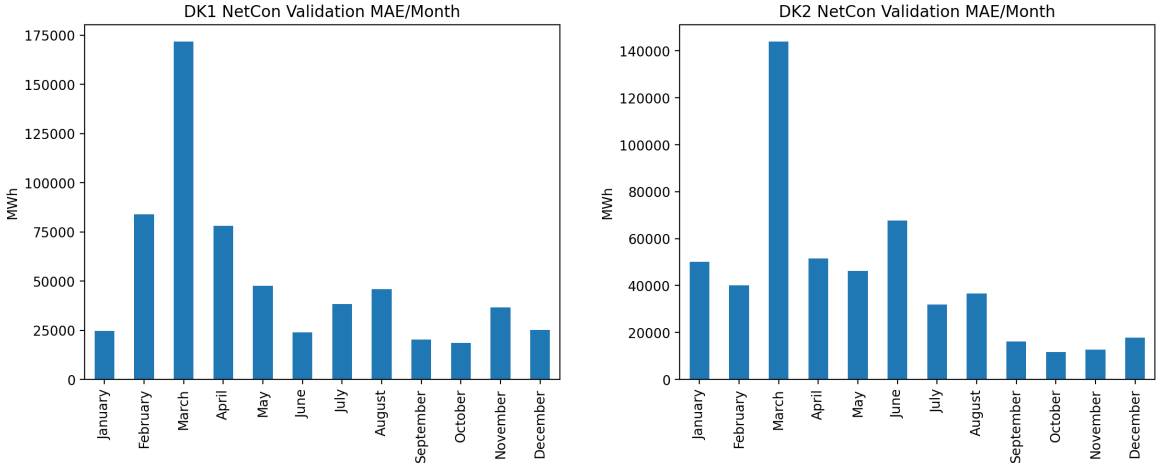
This suggested approach was applied to the consumption data at hand, but yielded less accurate predictions as measured by the MAE for both DK1 and DK2, when compared to the ARIMA models described in section 6.1.2 based on non-normalized data. As in the latter case, the automatically generated ARIMA models were again bested by the manually developed ones as measured by the MAE on the test data set. The prediction results of the manual models are shown in Figure 22.



(a) DK1: Test MAE = 51 241.08 MWh, Test MAPE = 3.07 %

(b) DK2: Test MAE = 43 842.57 MWh, Test MAPE = 4.16 %

Figure 22: Electricity consumption training (blue) and testing (green) data with predicted consumption (red) for the two areas DK1 and DK2. Predictions generated by manually developed seasonal ARIMA models based on workday-normalized consumption data.



(a) DK1

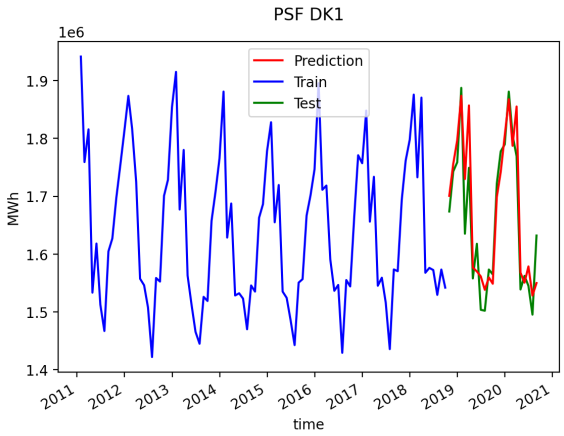
(b) DK2

Figure 23: Average MAE per month calculated on validation/test set forecasts. Predictions generated by manually developed seasonal ARIMA models based on workday-normalized consumption data.

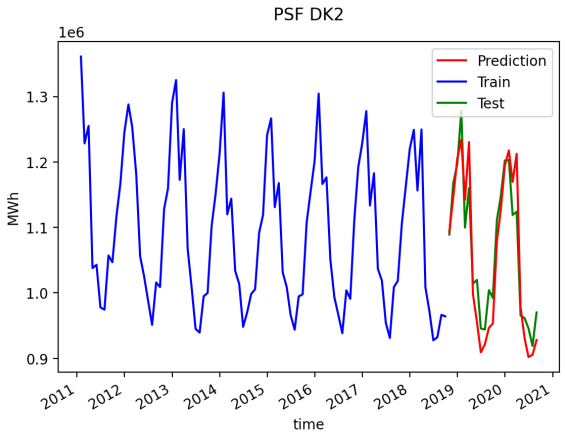
7.3 PSF

7.3.1 Original PSF

After the initial difficulties with the original PSF version using hourly data and a cycle length $c = 24$, i.e., one day, were alleviated by applying first order seasonal differencing, it was able to achieve more sensible results, which are shown in Figure 24.

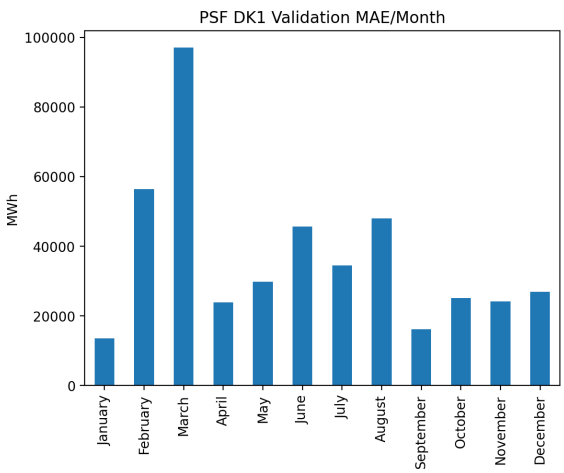


(a) DK1: Test MAE = 37 652.52 MWh, Test MAPE = 2.28 %

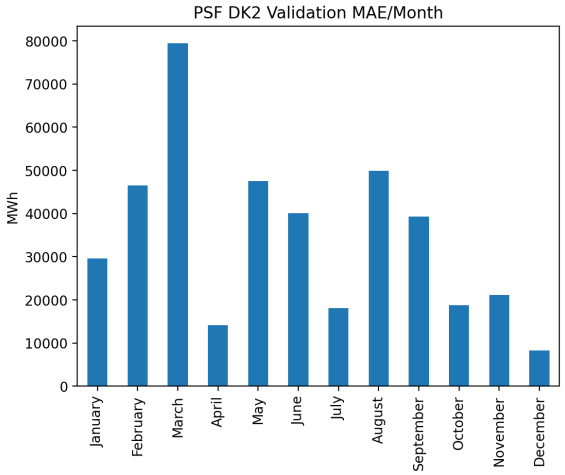


(b) DK2: Test MAE = 34 162.76 MWh, Test MAPE = 3.22 %

Figure 24: Electricity consumption training (blue) and testing (green) data with predicted consumption (red) for the two areas DK1 and DK2. Predictions generated using PSF with $c = 24$ on hourly electricity consumption data.



(a) DK1

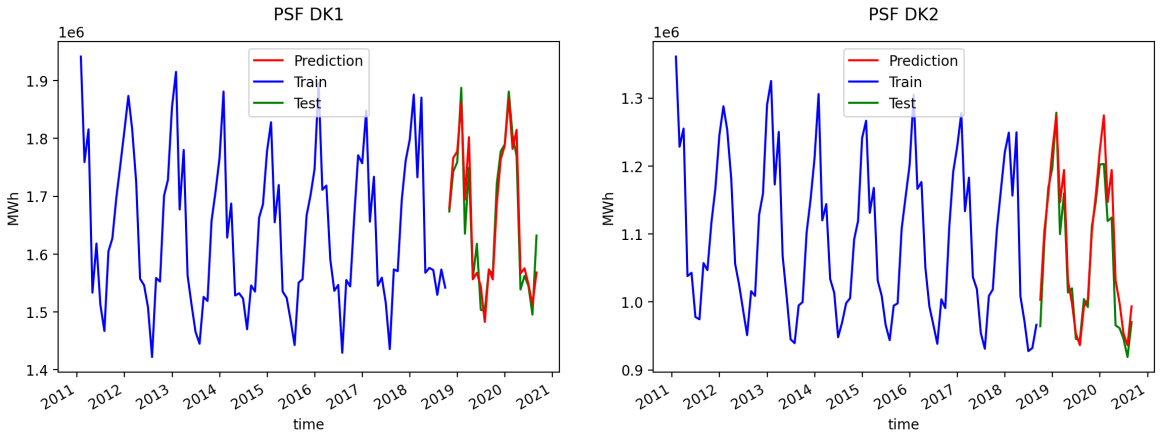


(b) DK2

Figure 25: Average MAE per month calculated on validation/test set forecasts generated by PSF with $c = 24$ on hourly electricity consumption data.

7.3.2 Year-long Cycle Length

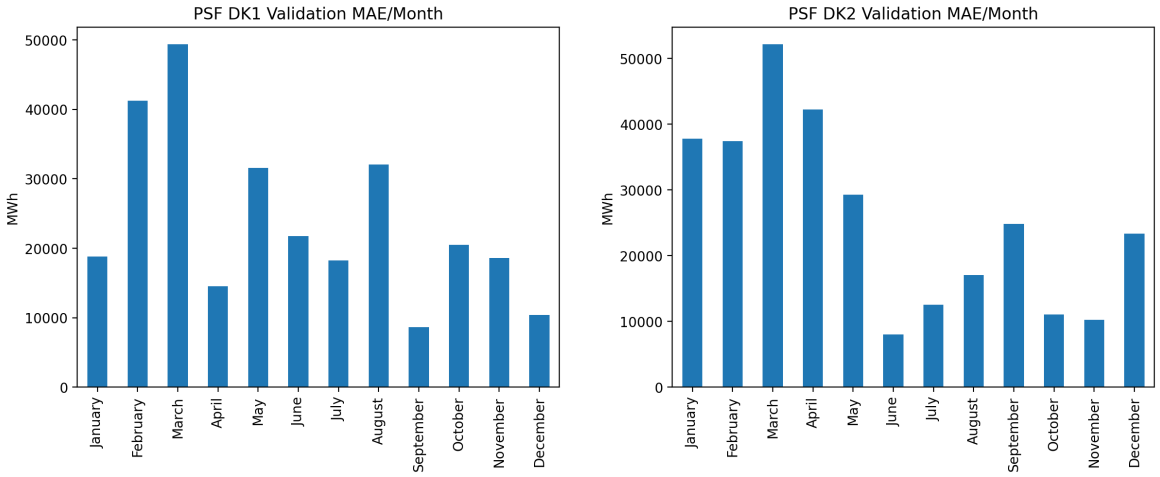
In the case of the consumption data from DK1, first summing the data to a daily time resolution, then running PSF to generate daily predictions and finally summing the predictions to the target monthly temporal resolution gave the lowest forecast error with a MAE of 24 458.57 MWh and MAPE of 1.47 % (compared to 28 723.71 MWh/1.66 % when directly predicting at a monthly time resolution). When running PSF on the consumption data from DK2, immediately summing the data to a monthly time resolution and subsequently directly generating monthly predictions gave the lower forecast error of 25 489.79 MWh or 2.4 % as measured by the MAE and MAPE, respectively. The associated forecast plots are shown in Figure 26.



(a) DK1: Test MAE = 24 458.57 MWh, Test MAPE = 1.47 %

(b) DK2: Test MAE = 25 489.79 MWh, Test MAPE = 2.4 %

Figure 26: Electricity consumption training (blue) and testing (green) data with predicted consumption (red) for the two areas DK1 and DK2. Predictions for DK1 were generated by PSF of daily values with $c = 365$ and then summing up to a monthly resolution, while the monthly predictions for DK2 were computed directly from monthly data with $c = 12$.



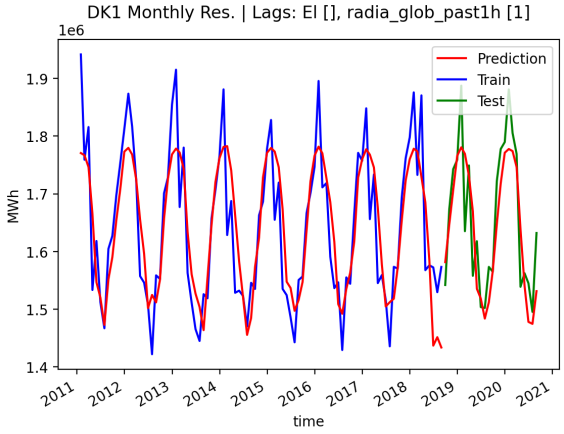
(a) DK1

(b) DK2

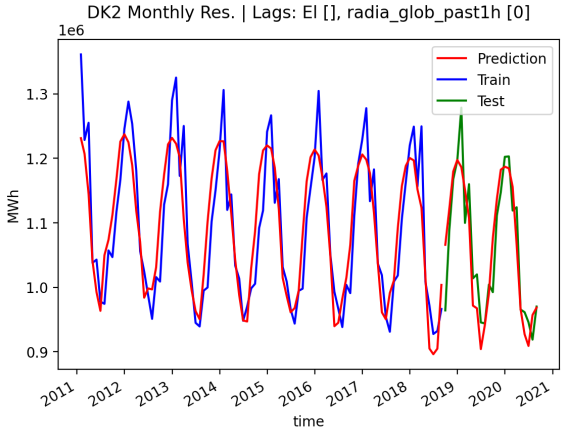
Figure 27: Average MAE per month calculated on validation/test set forecasts computed via PSF of daily values with $c = 365$ and then summing up to a monthly resolution for DK1, while the monthly predictions for DK2 were computed directly from monthly data with $c = 12$.

7.4 Multivariate Models

7.4.1 Linear Regression Using Single Weather Parameters

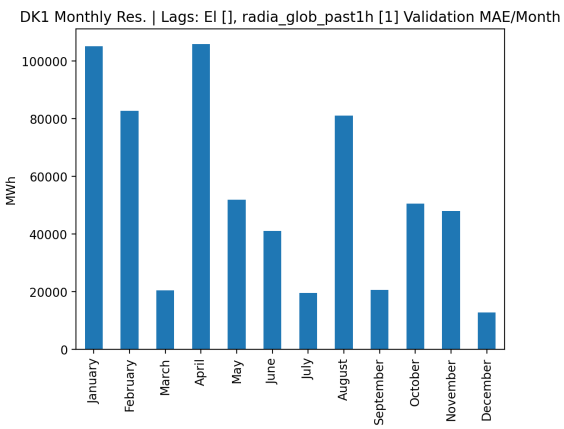


(a) DK1: Test MAE = 53 295.2 MWh, Test MAPE = 3.21 %

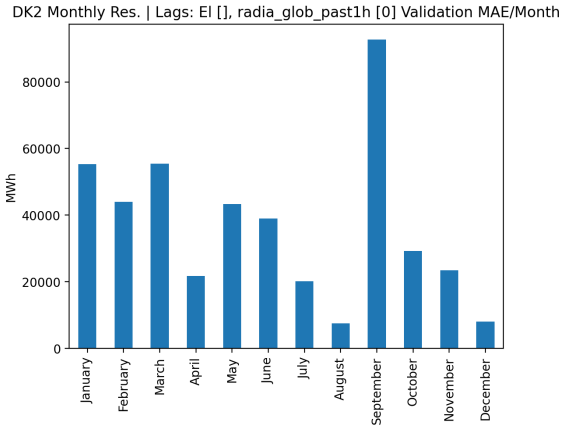


(b) DK2: Test MAE = 36 648.12 MWh, Test MAPE = 3.47 %

Figure 28: Electricity consumption training (blue) and testing (green) data with predicted consumption (red) for the two areas DK1 and DK2. Predictions generated by linear regression models with the two independent variables time and solar irradiance.



(a) DK1



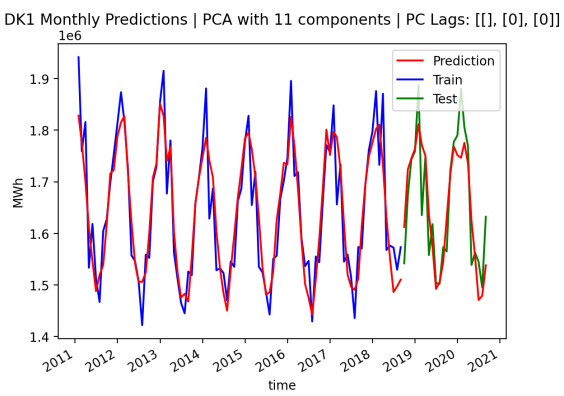
(b) DK2

Figure 29: Average MAE per month calculated on validation/test set forecasts. Predictions generated by linear regression models with the two independent variables time and solar irradiance.

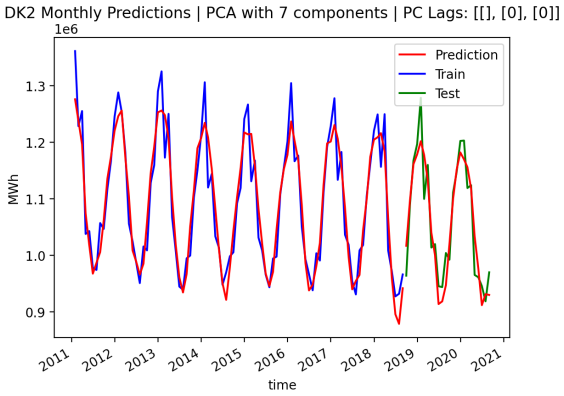
The MAE scores, as calculated on the test data set, of models (3) and (4) based on solar radiation were 53 295.2 MWh for DK1 and 36 648.12 MWh for DK2. The corresponding forecast plots are shown Figure 28.

7.4.2 Principal Component Regression Using Multiple Weather Parameters

The predictions of the principal component regression models that achieved the lowest MAE are shown in Figure 30. Both models used weather time series obtained through simple geospatial averaging.

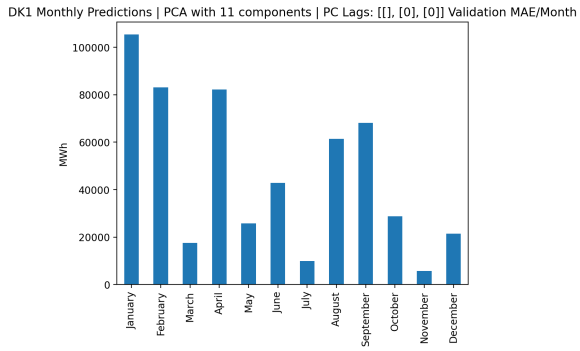


(a) DK1: Test MAE = 46007.27 MWh, Test MAPE = 2.78 %

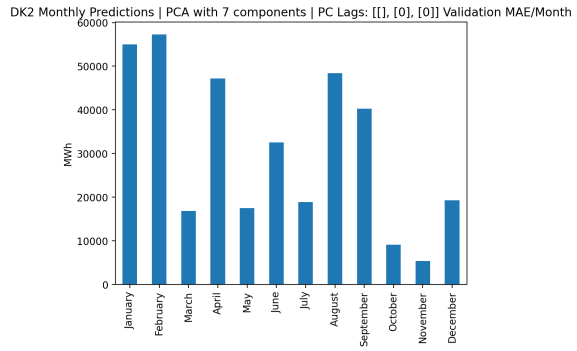


(b) DK2: Test MAE = 30645.75 MWh, Test MAPE = 2.92 %

Figure 30: Electricity consumption training (blue) and testing (green) data with predicted consumption (red) for the two areas DK1 and DK2. Predictions generated by multiple linear regression models using time and principal components as independent variables.



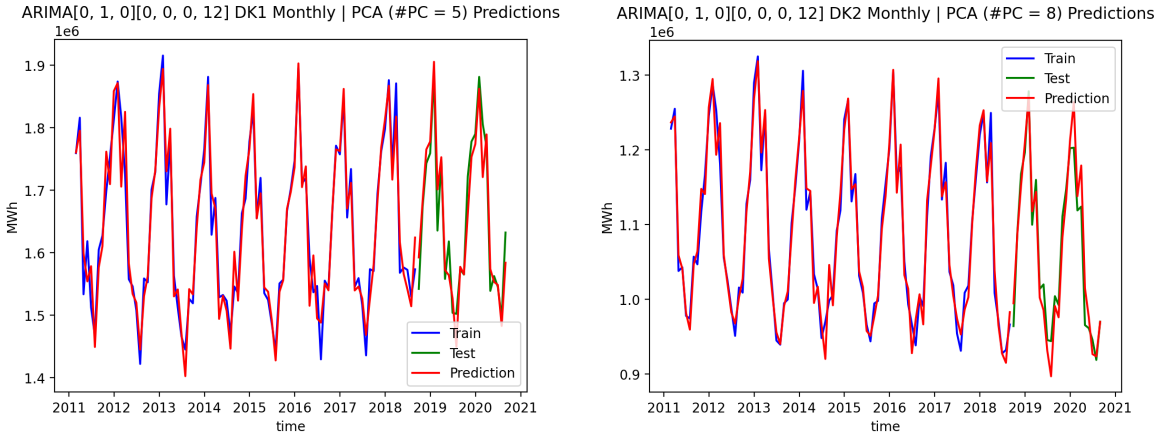
(a) DK1



(b) DK2

Figure 31: Average MAE per month calculated on validation/test set forecasts. Predictions generated by multiple linear regression models using time and principal components as independent variables.

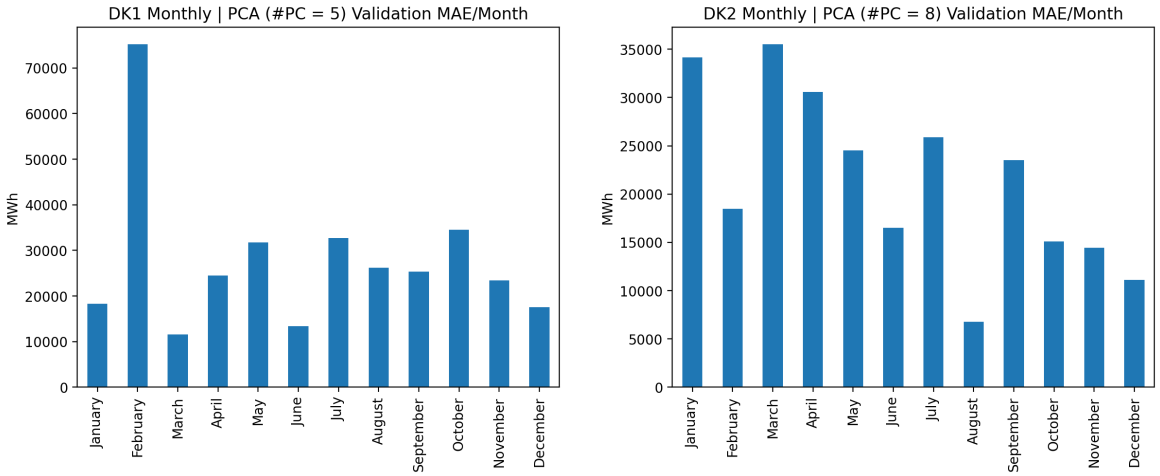
7.4.3 Combined Prediction Models



(a) DK1: Test MAE = 27 850.68 MWh, Test MAPE = 1.68 %

(b) DK2: Test MAE = 21 377.61 MWh, Test MAPE = 2.03 %

Figure 32: Electricity consumption training (blue) and testing (green) data with predicted consumption (red) for the two areas DK1 and DK2. Predictions were computed using combined PSF/PCR models with the lowest forecast error, based on monthly input data.



(a) DK1

(b) DK2

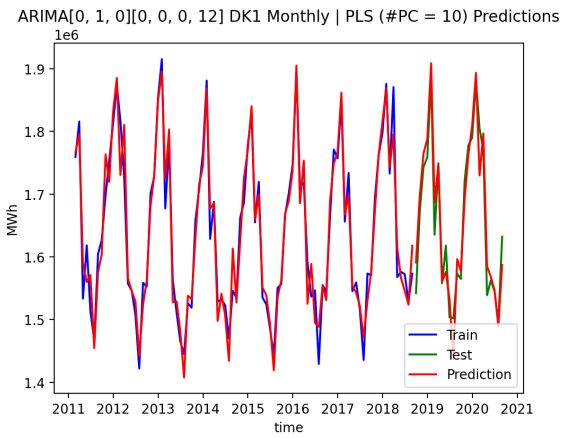
Figure 33: Average MAE per month calculated on validation/test set forecasts. Predictions were computed using combined PSF/PCR models with the lowest forecast error, based on monthly input data.

The combined model achieving the lowest MAE for price area DK1 used PLS for dimensionality reduction and, similar to best performing PCR model, Voronoi tessellation based geospatial averaging of the weather parameters. It achieved a test MAE of 27 419.02 and a MAPE of 1.66 % using 10 principal components (Shown in figure 34). Interestingly, another PLS based model instead using municipal geospatial averaging achieved a test MAE of 27 453.76 using only 3 principal components.

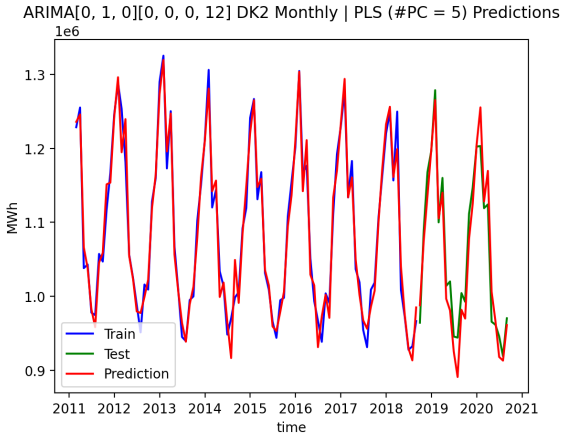
The PLS model that achieved the lowest prediction error for DK2 used weather data that was obtained through simple geospatial averaging. Again, the corresponding most accurate PCR model used geospatial averaging as well.

Figure 32 shows the forecast plots of the most accurate combined models based on PCA, while the

forecast plots of Figure 34 showcase the predictions of the most accurate combined models using PLS.

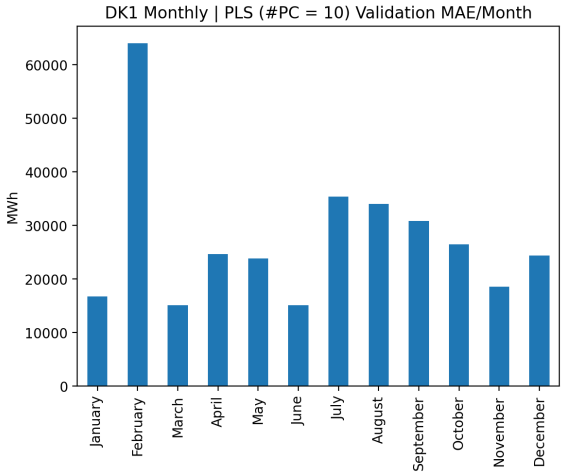


(a) DK1: Test MAE = 27 419.03 MWh, Test MAPE = 1.66 %

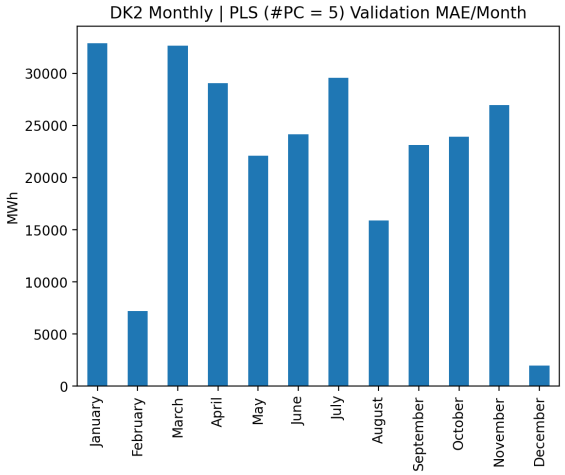


(b) DK2: Test MAE = 22 448.24 MWh, Test MAPE = 2.14 %

Figure 34: Electricity consumption training (blue) and testing (green) data with predicted consumption (red) for the two areas DK1 and DK2. Predictions were computed using combined PSF/PLS models with the lowest forecast error, based on monthly input data.



(a) DK1



(b) DK2

Figure 35: Average MAE per month calculated on validation/test set forecasts. Predictions were computed using combined PSF/PLS models with the lowest forecast error, based on monthly input data.

8 Discussion

Perhaps the most surprising result is the fact that it takes a considerable amount of effort to achieve a lower prediction error than the naive baseline models. Despite ARIMA being one of the most widely used univariate time series forecasting methods, both automatically generated ARIMA models failed to outperform the naive global average method, and even the manually developed ARIMA models were only able to beat it in the DK1 price area. The reason for the automatically generated DK1 model failing can probably be found in the lack of seasonal differencing, which together with a non-zero intercept c leads to the forecasts converging to the mean of the time series rather quickly [24].

Table 4: Comparing the test errors of the different prediction models. MAE is given in MWh and MAPE in percent. Table cells are shaded from lowest error (White) to highest (Black). Shading computed separately for DK1 and DK2.

Model	DK1 Test MAE	DK1 Test MAPE	DK2 Test MAE	DK2 Test MAPE
Naive (Average)	49950.94	2.9	27901.35	2.61
Naive (Repeat Prev.)	45334.24	2.59	32218.24	3.03
ARIMA (Auto)	63126.05	3.81	29464.94	2.75
ARIMA (Manual)	36918.42	2.23	28930.01	2.7
ARIMA (Workday)	51241.08	3.07	43842.57	4.16
PSF (Original)	37652.52	2.28	34162.76	3.22
PSF (Year Cycle)	24458.57	1.47	25489.79	2.4
Single Parameter Regr.	53295.2	3.21	36 648.12	3.47
PCR	46007.27	2.78	30645.75	2.92
Combined (PCA)	27850.68	1.68	21377.61	2.03
Combined (PLS)	27419.03	1.66	22448.24	2.14

The DK2 counterpart, which does apply seasonal differencing, is much closer to the observed series. Nonetheless, its prediction error is still higher than the best naive model. It appears that this is mainly due to a distinct double peak forecasting pattern, while also slightly underestimating the months of January and February. While the first peak is somewhat accurate, the second peak, corresponding to March, deviates considerably from the observed consumption in the test set. It is clear that this problem arises from the chosen model order only consisting of a single autoregressive term in the model. Since the estimated coefficient of that term is -0.044, each months prediction effectively corresponds to last years prediction after subtracting a small fixed amount (Due to seasonal differencing). From the forecast plot it is evident that this technique works well for the most part in the training set, but causes a problem when predicting out-of-sample data. This is due to the last year of the training set having a pronounced consumption peak in march, which does not occur in the two years of the test set.

We can see that the manually developed DK2 ARIMA model suffers from the exact same problem as its automatically generated sibling. However, it slightly alleviates the issue by having two seasonal AR terms, instead of only one non-seasonal AR term, and a seasonal MA term. Even the manually developed model for DK1 is affected by the double-peak of march 2018, despite including 4 non-seasonal AR and 5 seasonal AR terms. We can hence conclude that despite first order seasonal differencing being the right choice for this data, it had an unfortunate effect on the ARIMA predictions for both DK1 and DK2 in this particular instance, due the last year of training data differing from the majority of the previous years, while the test set again is more similar to the majority.

Using a single weather parameter in a least squares regression model, too, was not enough to rival the naive baseline models. With the single weather parameter giving the lowest test MAE being global solar irradiance, it is evident in the forecast plots that the model generally is more able to predict the consumption troughs than the peaks. An explanation for this could be that the power consumption in summer can to a larger degree be explained by the amount of solar radiation, while it is dominated more by temperature (among others) in winter, due to shorter days, lower solar altitude angle and the possibility of snow reflecting some of the sunlight. The forecast plot for DK2 also shows that not lagging the solar irradiance variable by one month has the effect in the training set that the predicted values preempt the observed values on the positive slopes (spring/summer) most of the time. In fact, the training MAE of the model using lagged solar irradiance is lower than that of the model included here. However, this discrepancy is reversed in the test set, where the model not using time lagging performs marginally better. Hence, it is possible that the DK2 model would generally perform better also using the time lagged version of solar irradiance, but due to happenstance did the opposite for this particular pair of test set years.

Involving all available weather parameters in PCR models had the desired effect of lowering the prediction error when compared to previous single weather parameter regression model, albeit still not enough to beat the naive baseline. The error per month bar plots show that the DK1 model improved in all months except for September and December, where it unexplainably worsened, and remained

almost equal in the months of January and February. The latter observation could point to the fact that the change in electricity consumption during these two months could not sufficiently be explained by just using weather parameters. However, it appears that the majority of prediction models were unable to achieve a low prediction error in February, increasing the likelihood that some other hard to predict factor influenced the consumption in this particular month of the test set. The DK2 PCR model improves upon the single weather parameter counterpart in most months as well, with only January remaining somewhat constant and April, August and December worsening. What causes this worsening in some months could not be determined.

Even the combined models, regardless of their dimensionality reduction method, were unable to significantly lower the prediction error in February for price area DK1, when compared to the other months. They were however, through the application of differencing, more sophisticated geospatial averaging and the inclusion of time indicator variables, able to improve the test prediction error by a wide margin over the regular PCR model. Both peaks and troughs are now approximated in much more detail and, disregarding the DK1 error peak of February, the prediction error is more uniformly distributed over the months. The fact that the combined prediction models produced ex-ante forecasts by first using PSF to compute weather forecasts for each weather parameter did not appear to considerably impact the prediction error. Preliminary analysis showed that using real weather observations to produce ex-post forecasts only reduced the forecast error in DK2, while increasing slightly in DK1. However, allowing the models to use future weather observations appeared to reduce the amount of principal components that needed to be retained.

The use of PLS for dimensionality reduction did not realize its theoretical advantage over PCA due to being supervised. It was only able to lower the prediction error in DK1, while being outperformed by PCA in DK2. It remains to be determined how this relationship changes when applying these models to a separate test set. In that case, it could be possible that one of PLS/PCA generalizes better in terms of retained principal components than the other.

Lastly, the alternative PSF method was able to achieve competitive results, both when compared to ARIMA and when compared to the combined prediction models, which also utilize PSF to forecast each weather parameter. Applying it naively to downsampled monthly electricity consumption data without consulting the validation set yielded results which almost matched the most accurate combined prediction model in DK1, while performing worse, but still better than any ARIMA model, in DK2. By allowing oneself to tune settings such as initial temporal resolution and differencing, the results could be improved further. It still remains unclear whether these results can be extrapolated to a separate test set, and hence one cannot conclude that PSF is a better model for forecasting electricity consumption in Denmark per se. For example, as discussed above, the unusual last year of the training electricity consumption data likely caused the ARIMA model to underperform, meaning that a different time period of data could cause the discrepancy in forecast error between PSF and ARIMA to shrink. On the other hand, another interpretation could be that PSF is more robust to the occurrence of such outliers in the training data.

A final notable aspect was that in all but one case, the non-naive models always achieved a lower validation MAPE when applied to DK1 data, when compared to DK2. They were also able to lower the relative prediction error for DK1 significantly more, despite the relatively constant error spike in the month of February. This opens up the question whether there exist some factors inherent to the electricity consumption in DK2 that causes it to be more unpredictable than the consumption in DK1. However, before looking to answer that question, it would be wise to first investigate whether this phenomenon is also present in the new "Production and Consumption" data set that was released by Energinet in March 2021.

8.1 Limitations

Without the inclusion of a separate test set to further evaluate how these results generalize to unseen data, only limited conclusions can be drawn. All but the automatically generated ARIMA models were tuned on a validation set, be it by choosing the optimal temporal resolution for prediction in the case of PSF, changing the amount of autoregressive and moving average terms in the ARIMA models or choosing the number of principal components to retain in the combined prediction models.

Whether the forecast accuracy obtained through this tuning can be extrapolated to other/future data can not be said with certainty. It would have been interesting to test the selection of best models selected based on the validation error on a final test set. Unfortunately, the electricity consumption data set was deprecated and replaced by another dataset on the 1. February 2021. This data set no longer includes net consumption explicitly and trying to compute it by subtracting the transmission losses from the gross consumption results in data that differs from the original data set. In case of the original data set, only using the data from the 1. September 2020 to the 1. February was deemed as not sufficient for a final test data set, as this last section of the data differs significantly from all historical data, especially in the case of DK1, leading to all models, naive or not, failing completely to predict it accurately. This discrepancy can not be found in the new "Production and Consumption" dataset that replaced the original dataset and is therefore assumed to be erroneous.

Regarding the combined prediction models, the use of a relatively crude weather forecasting method when making real-world forecasts also introduces uncertainty and may affect the optimal number of principal components to retain. Both PCA and PLS appear to be very sensitive to changes in the data, e.g., when a different weather forecast is used in the input. They are also sensitive to outliers, some of which still remain even after cleaning the data. This sensitivity can make it difficult to extrapolate the results found for this data set to others, or even future data. In addition, the effects of the pandemic on electricity consumption have not been accounted for in any way. This could have influenced the test errors of all models negatively.

On the other hand, the forecasts computed by the simple regression models using single weather parameters and PCA regression using multiple weather parameters are all ex-post, i.e., they all use real weather observations for their out-of-sample predictions. Hence, their prediction performance may be more accurate than it would be in the real world, where ex-ante forecasts must be utilized.

Lastly, it should be noted that all forecasts are provided without confidence intervals, which could potentially induce a false sense of security for forecasts that lie far in the future. However, this is somewhat mitigated by the fact that electricity consumption time series follows a seasonal pattern, meaning that it is less like to suddenly deviate far from its usual range of values, as opposed to, e.g., stock market price time series.

8.2 Suggestions for Further Research

1. An point for future work could be to evaluate the effect of the SARS-CoV-2 pandemic on electricity consumption in Denmark. This would open up the possibility to account for these effects when training future models, or re-evaluate the models developed in this thesis on data that is not tainted by the effects of a pandemic.
2. This project used data from the "metObsAPI" for raw weather observations, which have to be cleaned manually cleaned and thus have a negative effect on the electricity consumption forecast accuracy that any model can achieve. In 2021, DMI is to make a new climate data API available to the public, which will provide quality controlled weather observations. These data should probably be preferred in any future projects, unless data cleaning is part of the problem formulation. With quality controlled data, it might be possible to construct models of higher spatial resolution. It would, for example, be possible to develop one model for each weather station, and then combine these predictions to obtain more accurate country-wide predictions. To this end, it would perhaps be beneficial to re-use the Voronoi diagram method to obtain a per-station electricity consumption estimate.
3. By applying non-linear methods, including artificial neural networks, on could determine if they are able to produce more accurate predictions than the models discussed in this thesis. On top of regular and perhaps deep Feed-Forward Neural Networks (FFNNs), it would be interesting to apply recurrent neural networks such as Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM), since they have been shown to be promising methods for this particular problem in recent papers [15]. LSTMs have also been used in conjunction with attention models [11] and dropout regularization techniques [17] to improve prediction accuracy and interpretability. Convolutional Neural Networks (CNNs) have also been applied to time series forecasting, and have even been combined with LSTMs, which could make them interesting candidates investigate

in the context of electricity consumption forecasting [29]. The PSF method that was applied in this thesis has also been combined with neural networks in the past [30], so it could also be interesting to further research in this direction. Some researches have also found fuzzy neural networks to work well for demand forecasting [35]. These more complex models might also be able to extract more information from high temporal resolution hourly or daily data, and thus pave the way for more detailed data augmentation, such as adding variables that indicate holidays or agricultural irrigation.

4. While electricity consumption forecasting has historically mostly comprised discrete single-valued prediction, more recently probabilistic prediction models have gained interest in the research community [22] [20] [21]. They provide a prediction for any given time step in terms of a number of quantiles, representing the range of values the target variable might take in terms of probability. It might be interesting to investigate whether this approach is more useful than the traditional point forecasting methodology.

5. Instead of the ex-ante forecasts produced by the combined prediction models relying on PSF to forecast the weather parameters, it might be preferable to instead compute multiple forecasts, driven by a number of possible scenarios. For example, one might use data from the coldest or hottest years to provide an upper/lower bound on the future expected electricity demand. It could also be worthwhile to include confidence intervals in the forecasts of complex models, to achieve an effect similar to probabilistic forecast methods.

6. This project used a single validation data set for model selection, partly because only nine years of electricity consumption data was available. In the future, it might be worthwhile to instead use cross-validation methods for model selection, assuming enough data is available. In this context, the "Production and Consumption" dataset that was released by Energinet in March of 2021 might be interesting. It contains electricity consumption data starting in the year 2005, and as such might be a good candidate for cross-validation. Further, the extra amount of data allows to hold out a test set while still retaining more training data than was contained in the dataset used in this project. This could also improve the prediction quality of weather parameter based machine learning models, as they would have a better opportunity to establish the relationship between the weather parameters and electricity consumption.

7. Since the relationship between weather parameters, especially temperature, and electricity consumption flattens out towards the highest weather measurements (See Figure 11), an effect that is even stronger for higher temporal resolutions, it may be that prediction models relying solely on external weather variables are less accurate in the summer than in the winter. In this case, it could be worthwhile to determine if there exists a univariate prediction model that is able to outperform the weather based one in the summer months. Provided such a model is found, it would make sense to develop a hybrid model that produces a weighted average of the weather based and the univariate model, where the weights gradually change such that winter forecasts are 100 % given by the former, while summer forecasts are 100 % given by the latter.

9 Conclusion

This project aimed to investigate the relationship between weather parameters and net electricity consumption in Denmark and determine if it could be utilized to predict future electricity consumption via machine learning. Linear correlation analysis of the cleaned weather and electricity demand time series revealed that at least 8 out the 11 weather parameters were sufficiently highly correlated to be used as external variables in prediction models. These models ranged from simple regression models using individual weather variables to dimensionality reduction based multiple regression employing data augmentation and differencing in an attempt to model the dynamic time aspect of the prediction problem. ARIMA and PSF, univariate machine learning models, were utilized to examine if the inclusion of weather parameters was a necessity to achieve low forecast errors, in addition to enabling ex-ante forecasts to be made. Two naive prediction methods were implemented to serve as baseline models.

The results suggest that weather parameters can advantagously be included in regression-type pre-

diction models to lower the prediction error. While the univariate ARIMA models were not able to significantly outperform their naive counterparts, the PCA/PLS based multiple regression models using integrated weather variables were able to do so in both electricity zones, provided that the optimal number of principal components was retained. The results also suggest that other prediction methods not involving regression such as PSF could be competitive alternatives, as well as having the advantage of not requiring feature engineering and model tuning. More research is needed on applying these alternative models to the specific problem of electricity consumption forecasting. Similarly, future work is needed to determine if FFNNs and the newer RNNs are able to outperform regression-type and alternative methods in the univariate and multivariate cases, especially when applied to high temporal/spatial resolution data, where the extra complexity of these models could potentially be leveraged to learn more detailed relationships. Additionally, it remains to be determined if the optimal number of retained principal components w.r.t. the validation prediction error also performs well on a separate test set.

In general, conclusions drawn from case studies such as this are subject to the caveat of uncertainty, which arises from limitations such as scope, sample size and data quality. In this case, the models were only applied to data from Denmark, and no final test set was used due to only a limited amount of data being available. Further, although efforts where made to exclude as many erroneous measurements as possible, not all could successfully be removed from the data, thus influencing the results of all weather parameter based models. Therefore, it is not possible to point out one machine learning model in particular as the best. Rather, one may view the results of this project as indications suggesting which subset of models could be promising candidates for further research based on their validation set forecast error.

References

[1] auto_arima function documentation. https://web.archive.org/web/20201114193854/https://alkaline-ml.com/pmdarima/modules/generated/pmdarima.arima.auto_arima.html?highlight=auto_arima. Archived: 14.11.2020.

[2] DMI parameters (metobs) table. https://web.archive.org/web/20201110204942/https://confluence.govcloud.dk/pages/viewpage.action?pageId=26476616. Archived: 10.11.2020.

[3] DMI vejrekstremer i danmark. https://www.dmi.dk/vejrarkiv/vejrekstremer-danmark/.

[4] geovoronoi documentation. https://web.archive.org/web/20200505133605/https://pypi.org/project/geovoronoi/. Archived: 05.04.2020.

[5] Interactive plotly geographic plot of DMI weather stations. https://web.archive.org/web/20201110221000/https://mamei16.github.io/. Archived: 10.11.2020.

[6] Newport: Introduction to solar radiation. https://web.archive.org/web/20210418012914/https://www.newport.com/t/introduction-to-solar-radiation. Archived: 18.04.2021.

[7] Scipy voronoi function documentation. https://web.archive.org/web/20201112042639/https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.Voronoi.html. Archived: 12.11.2020.

[8] TV2: Her er de danske vejrrekorder fra de seneste 10 år. https://web.archive.org/web/20191230162055/https://vejr.tv2.dk/2019-12-28-her-er-de-danske-vejrrekorder-fra-de-seneste-10-aar. Archived: 30.12.2019.

[9] Wikipedia: Atmospheric pressure. https://web.archive.org/web/20210502031531/https://en.wikipedia.org/wiki/Atmospheric_pressure. Archived: 02.05.2021.

[10] Wikipedia: Visibility. https://web.archive.org/web/20210211234743/http://en.wikipedia.org/wiki/Visibility. Archived: 11.02.2021.

[11] Hossein Abbasimehr and Reza Paki. Improving time series forecasting using lstm and attention models. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–19, 2021.

[12] Francisco Martinez Alvarez, Alicia Troncoso, José C Riquelme, and Jesus S Aguilar Ruiz. Energy time series forecasting based on pattern sequence similarity. *IEEE Transactions on Knowledge and Data Engineering*, 23(8):1230–1243, 2010.

[13] Sabyasachi Basu and Martin Meckesheimer. Automatic outlier detection for time series: an application to sensor data. *Knowledge and information systems*, 11(2):137–154, 2006;2007;.

[14] Neeraj Bokde, Gualberto Asencio-Cortés, Francisco Martínez-Álvarez, and Kishore Kulat. Psf: Introduction to r package for pattern sequence based forecasting algorithm. *arXiv preprint arXiv:1606.05492*, 2016.

[15] Salah Bouktif, Ali Fiaz, Ali Ouni, and Mohamed Adel Serhani. Optimal deep learning lstm model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches. *Energies*, 11(7):1636, 2018.

[16] M Davies. The relationship between weather and electricity demand. *Proceedings of the IEE-Part C: Monographs*, 106(9):27–37, 1958.

[17] Benjamin Donnot, Isabelle Guyon, Marc Schoenauer, Antoine Marot, and Patrick Panciatici. Fast power system security analysis with guided dropout. *arXiv preprint arXiv:1801.09870*, 2018.

[18] Jean Gallier. *Dirichlet-Voronoi Diagrams and Delaunay Triangulations*, pages 267–286. Springer New York, New York, NY, 2001.

[19] L. Hernandez, C. Baladron, J. M. Aguiar, B. Carro, A. J. Sanchez-Esguevillas, J. Lloret, and J. Massana. A survey on electric power demand forecasting: Future trends in smart grids, microgrids and smart buildings. *IEEE Communications Surveys Tutorials*, 16(3):1460–1495, 2014.

[20] Tao Hong and Shu Fan. Probabilistic electric load forecasting: A tutorial review. *International Journal of Forecasting*, 32(3):914–938, 2016.

[21] Tao Hong, Pierre Pinson, Shu Fan, Hamidreza Zareipour, Alberto Troccoli, and Rob J Hyndman. Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond, 2016.

[22] Tao Hong, Jason Wilson, and Jingrui Xie. Long term probabilistic load forecasting and normalization with hourly information. *IEEE Transactions on Smart Grid*, 5(1):456–462, 2013.

[23] Ching-Lai Hor, Simon J Watson, and Shanti Majithia. Analyzing the impact of weather variables on monthly electricity demand. *IEEE transactions on power systems*, 20(4):2078–2085, 2005.

[24] R.J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice, 2nd edition*. OTexts: Melbourne, Australia, 2018. Accessed on: 30.05.2021.

[25] Rob J. Hyndman and Yeasmin Khandakar. Automatic time series forecasting: The forecast package for r. *Journal of statistical software*, 27(3):1–22, 2008.

[26] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

[27] Fazil Kaytez, M. Cengiz Taplamacioglu, Ertugrul Cam, and Firat Hardalac. Forecasting electricity consumption: A comparison of regression analysis, neural networks and least squares support vector machines. *International Journal of Electrical Power & Energy Systems*, 67:431 – 438, 2015.

[28] Denis Kwiatkowski, Peter C. B. Phillips, Peter Schmidt, and Yongcheol Shin. Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of econometrics*, 54(1):159–178, 1992.

[29] Martin Längkvist, Lars Karlsson, and Amy Loutfi. A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters*, 42:11–24, 2014.

[30] Mostafa Majidpour, Charlie Qiu, Peter Chu, Rajit Gadh, and Hemanshu R Pota. Modified pattern sequence-based forecasting for electric vehicle charging stations. In *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 710–715. IEEE, 2014.

[31] Francisco Martínez-Álvarez, Alicia Troncoso, José C Riquelme, and Jesús S Aguilar-Ruiz. Lbf: A labeled-based forecasting algorithm and its application to electricity price time series. In *2008 Eighth IEEE International Conference on Data Mining*, pages 453–461. IEEE, 2008.

[32] S Mirasgedis, Y Sarafidis, E Georgopoulou, DP Lalas, M Moschovits, F Karagiannis, and D Papakonstantinou. Models for mid-term electricity demand forecasting incorporating weather influences. *Energy*, 31(2-3):208–227, 2006.

[33] A Pankratz. *Forecasting with dynamic regression models*. New York: John Wiley, 1991.

[34] Allan Pinkus. Approximation theory of the mlp model in neural networks. *Acta Numerica*, 8:143–195, 1999.

[35] Henrique Pombeiro, Rodolfo Santos, Paulo Carreira, Carlos Silva, and João MC Sousa. Comparative assessment of low-complexity models to predict electricity consumption in an institutional building: Linear regression vs. fuzzy modeling vs. neural networks. *Energy and Buildings*, 146:141–151, 2017.

[36] Henrique Pombeiro, Rodolfo Santos, Paulo Carreira, Carlos Silva, and João M.C. Sousa. Comparative assessment of low-complexity models to predict electricity consumption in an institutional building: Linear regression vs. fuzzy modeling vs. neural networks. *Energy and Buildings*, 146:141 – 151, 2017.