Time Series Forecasting for Residential Energy Consumption with ARIMA and SVR

Machine Learning mit Anwendung in der Ökonomie
Max Gaber (maxgaber@uni-bremen.de), Universiät Bremen
July 22, 2020

Abstract — **Matching power supply and demand is crucial for the stability of the energy system. Intelligent forecasting methods can help to sustain stability and efficiency and support decision making regarding infrastructure and resources. In this paper, the Machine Learning models ARIMA and SVR are applied on a uni-variate load time series for a single household and tested for their applicability. As the time series contains a dominant seasonal pattern from which it only deviates randomly, it is observed that even trivial models such as ARIMA can compete with more complex methods. The results show a similar prediction accuracy for both models, that are both able to forecast the seasonal component, but failing on the non-seasonal patterns.**

1 Introduction

Energy systems undergo a fundamental transformation nowadays, moving from a primarily fossil energy production towards a regenerative supply in order to contribute to a more carbon neutral economy (Albert et al. 2016). In contrast to the traditional production, renewable energies cannot be adjusted to the actual energy demand as they often rely on external factors such as sun or wind. Matching power supply and demand is, however, crucial for the stability of the energy system.

In times of growing energy demand in conjunction with increasing renewable shares, forecasting the load is a promising way to sustain stability and efficiency in an intelligent way – the so-called Smart Grid. It helps to allocate the necessary resources such as storage units or balancing power and supports decision making regarding infrastructure, investments, maintenance and pricing – just to mention a few areas of application.

As the residential energy consumption makes up 27% of the total energy consumption worldwide (Nejat et al. 2015), forecasting and modeling this sector's load is a cornerstone of understanding the use of energy better and to allocate the necessary resources to improve the efficiency of the network. However, predicting energy consumption on a building level is complex due to the many factors influencing the consumption. Weather, technical facilities and occupation patterns all play a role in the actual household's energy load.

Thanks to the spreading usage of smart meters, nowadays, a significant amount of data is available for research on consumption patterns and engineering of forecasting systems. Machine Learning techniques have proven their ability of predicting time series accurately based on historical data. Deb et al. 2017 provide a comprehensive overview of the state-of-the-art Machine Learning techniques at the time and their respective accuracy when applied for energy load prediction.

The aim of this work is the evaluation of different Machine Learning techniques on a given energy consumption dataset based on uni-variate time-series prediction. The chosen data for this study is the "UCI Individual household electric power consumption dataset" (Hebrail and Berard 2012) and will be referenced as IHEPC data set hereafter. It contains electricity load data for a single household in a time period of several months. It is intended to apply an ARIMA model for forecasting and thereby reproduce an already conducted study as a reference first. After that, an independently implemented approach with the Support Vector Regression (SVR) method is going to be tested on the same data and compared with the previous results and with reference literature on the IHEPC dataset.

In the following, the paper is going to outline the related scientific literature in section 2 first, then focus on the theory behind ARIMA and SVR (Section 3), the specifics of the IHEPC dataset (Section 4), the methodical approach (Section 5) and finally give an evaluation in section 6 on the results obtained.

2 Literature Review

A variety of research exists for the IHEPC dataset, covering simple statistical forecasting techniques like regression models as well as more sophisticated approaches like Artificial Neural Networks or Long Short-term Memory Networks.

Applying a statistical model, Chujai et al. 2013 use the AutoRegressive Moving Average (ARMA) method and the AutoRegressive Integrated Moving Average (ARIMA) method for medium-term prediction up to one year, respectively with a daily, weekly, monthly and quarterly periodicity. Also, Beliaeva et al. 2013 investigate the use of ARIMA on daily and weekly forecasts, involving the Box-and-Jenkins method to determine the model configuration.

Puspita and Ermatita 2019 investigate the applicability of Support Vector Machines (SVM) in combination with Principal Component Analysis (PCA) for short-term forecasts up to 30 min ahead.

However, many papers refer to techniques using different kinds of Deep Learning. A Long Short-term Memory Recurrent Neural Network (LSTM) is proposed by Marino et al. 2016 in a standard and in a sequence-to-sequence configuration, obtaining convincing results for the latter with minutely and hourly frequency for a one-year forecasting horizon. With a similar setup, Mocanu et al. 2016 use the Conditional Restricted Boltzmann Machine (CRBM) and the Factored Conditional Restricted Boltzmann Machine (FCRBM), which outperform Artificial Neural Networks (ANN) and SVMs in the experiment.

Multiple authors focus in their work on Convolutional Neural Networks (CNN) in combination with LSTM. Khan et al. 2020 use it for a precise prediction on an hourly, daily and weekly basis. With a multivariate sliding window approach, Kim and Cho 2019 propose a CNN-LSTM that outperforms linear regression, random forest, decision tree and multilayer perceptrons in their study, considering minutely, daily and weekly resolutions. Combined with a bi-directional LSTM, Le et al. 2019 achieve an out-performance of the CNN in comparison with ordinary CNN-LSTMs, standard LSTMs and linear regression with three years learning and two years testing scopes.

3 Theoretical Background

ARMA and ARIMA Model

ARMA and ARIMA are statistical models often applied for time series modeling or forecasting. Their acronym stands for *Autoregressive Moving Average* or *Autoregressive Integrated Moving Average* respectively and implies a combination of different modeling strategies. The major goal is predicting future time series values z_t at the time t based on a limited amount of past data $z_{\mathrm{t-x}}$ through different regressive methods.

ARMA thus consists of two major parts: the AR-term and the MA-term. The autoregressive term derives a temporal value from its preceding values through a regression while the moving average term considers the previous residuals instead. In order to predict the value

$\bar{z}_{\mathrm{t}}$ at a given time t, equation (1) describes the relation as a sum of AR and MA terms.

$$\bar{z}_{\mathrm{t}} = \sum_{\mathrm{i}=1}^{p} \Phi_{\mathrm{i}} z_{\mathrm{t-i}} - \sum_{\mathrm{j}=1}^{q} \Theta_{\mathrm{j}} a_{\mathrm{t-j}} + a_{\mathrm{t}} \tag{1}$$

Here, $\bar{z}_{\mathrm{t}} = z_{\mathrm{t}} - \mu$ is the deviation of z_{t} from some origin μ (e.g. the mean) and Φ_{i} and Θ_{j} are the parameters of the auto-regressive and moving-average part respectively. The term a_{t} stands for the residual at the observed time t and describes white noise or rather a random residual. It is required that the time series is stationary in its mean and variance.

ARMA thus holds 2 hyperparameters to be tuned and is often denoted as ARMA(p, q): p is the number of lagged values $z_{\mathrm{t-i}}$ to be considered as features in the regression process while q is the number of past error terms $a_{\mathrm{t-j}}$ used for the moving average – also known as the averaging window size (Makridakis and Hibon 1997).

Using the backshift operator $\hat{B}^{(j)}$, the governing equation can be denoted as in equation (2), holding $\hat{B}^{(j)} z_{\mathrm{t}} = z_{\mathrm{t-j}}$.

$$\Phi(\hat{B})\bar{z}_{\mathrm{t}} = \Theta(\hat{B}) a_{\mathrm{t}} \tag{2}$$

Thus, $\Phi(\hat{B})$ and $\Phi(\hat{B})$ (Equations 3 and 4) are the polynomial equations that already can be observed in equation (1), now, however, described by the backshift operator. (Refer for the complete derivation to Box et al. 2008).

$$\Phi(\hat{B}) = 1 - \Phi_1 \hat{B} - \Phi_2 \hat{B}^{(2)} - ... - \Phi_{\mathrm{p}} \hat{B}^{(p)} \tag{3}$$

$$\Theta(\hat{B}) = 1 - \Theta_1 \hat{B} - \Theta_2 \hat{B}^{(2)} - ... - \Theta_{\mathrm{q}} \hat{B}^{(q)} \tag{4}$$

Using this mathematical approach, time series modeling is feasible after considering some preparational steps described in Box et al. 2008 and named as the *Box-and-Jenkins-Methodology*. First, it needs to be assured that the time series is stationary. If that is not the case, stationarity can be achieved by deriving the time series to the d^{th} order – a method for which the *I* in *ARIMA* stands. d is another hyperparameter that needs to be tuned. Then, seasonality needs to be determined: If the times series data contains seasonal patterns, a seasonal ARIMA$(p, d, q)(P, D, Q)_{\mathrm{m}}$ approach can be applied, which models the seasonality separately. P, D and Q denote the seasonal parameters while m stands for the seasonal period length. The seasonality is described by the seasonal backshift operator $\hat{B}_{\mathrm{S}}$ and implies a multiplicative relation for each of the AR- and MA-terms in equation 5.

$$\phi(\hat{B}_{\mathrm{S}})\Phi(\hat{B})\bar{z}_{\mathrm{t}} = \theta(\hat{B}_{\mathrm{S}})\Theta(\hat{B}) a_{\mathrm{t}} \tag{5}$$

As a subsequent step, Φ and Θ need to be determined. This can be achieved by a non-linear optimization procedure on the in-sample values, e.g. Gradient Descent. In the end, before applying the model, the results should be carefully diagnosed to ensure the model's adequacy. It is necessary, that the residuals of the result are random (white noise) and do not contain any relevant information on the underlying relations.

p, d, q and eventually P, D and Q are parameters unknown in the beginning. There exist several ways for the determination of those hyperparameters of which the grid search is chosen in this study, considering the Akaike Information Criterion (AIC) as suggested by Brockwell and Davis 2016. The AIC is calculated on the in-sample set and acts as an indicator for the quality of the fit and an estimator for the out-of-sample error (see equation 6). The lower the AIC is, the better performs the model with the corresponding selection of hyperparameters. The AIC consists on the one hand of an estimator for the goodness of the fit, which is represented by the maximum logarithmic likelihood function L, measuring the probability density that a given set of data might occur in the distribution of the training data. To prevent the model from overfitting, a punishing term is implemented on the other hand, considering the number of parameters k for the sake of parsimony.

$$\text{AIC} = 2k - 2\ln(\hat{L}) \tag{6}$$

Box et al. 2008 thus provide a convenient work flow for model determination, which is going to be followed also in this study. It should be acknowledged however, that the effectiveness of this approach is a subject of controversy, e.g. in Commandeur and Koopman 2007.

Support Vector Regression

Although Support Vector Machines (SVM) originally have been used for classification tasks, they can be applied for regression as well. This section introduces non-exhaustively the key points of their theoretical background, referring to Cristianini and Shawe-Taylor 2000 for a complete derivation, which acts as literature source for this chapter too.

SVM – and in this case Support Vector Regression (SVR) specifically – is a linear machine learning method in a kernel-induced feature space, by solving the equation $\hat{y}_\mathrm{i} = \omega^T x_\mathrm{i} + b$ as a linear regression problem. $\hat{y}_\mathrm{i}$ is the predicted value, x_i the corresponding feature vector and ω and b the coefficient matrix. The optimization problem is solved by minimizing the cost function J (Equation 7) by finding the optimal set of parameters ω. The cost function is regularized for large parameter weights in the first part to avoid overfitting and for errors or misclassifications in the second term.

$$J = \frac{1}{2}||\omega||^2 + C\sum_{i=1}^{N} L_\epsilon(y_\mathrm{i}, \hat{y}_\mathrm{i}) \tag{7}$$

$$L_\epsilon = \begin{cases} 0 & \text{if } |y_\mathrm{i} - \hat{y}_\mathrm{i}| \leq \epsilon \\ |y_\mathrm{i} - \hat{y}_\mathrm{i}| - \epsilon & \text{else} \end{cases} \tag{8}$$

Characteristic for SVRs is the ϵ-intensive loss function L_ϵ – the ignorance of deviations within an error range of $\pm\epsilon$ from the regression hyper plane, only considering values lying outside a tolerance corridor, which is reflected in the second term using equation 8, $\hat{y}_\mathrm{i}$ denoting the estimate value. Those values contribute with an error of ξ_i (above ϵ) and ξ_i^* (below ϵ) to the cost function and are regulated by the hyperparameter C, which needs to be tuned in the model selection process. The final cost function is given in equation 9.

$$J = \frac{1}{2}||\omega||^2 + C\sum_{i=1}^{N}(\xi_\mathrm{i} + \xi_\mathrm{i}^*) \tag{9}$$

Minimizing the cost function in order to retrieve the optimal set of parameters follows the constraints being set to the error values in equation 10. ϵ and C are problem specific constants and need to be adjusted taking the application's specifics into account.

$$\begin{gathered} y_\mathrm{i} - (\omega^T x_\mathrm{i} - b) \leq \epsilon + \xi \\ -y_\mathrm{i} + (\omega^T x_\mathrm{i} + b) \leq \epsilon + \xi_\mathrm{i}^* \\ \xi_\mathrm{i}, \xi_\mathrm{i}^*, C \geq 0 \end{gathered} \tag{10}$$

The minimization of J is a typical optimization problem and can be achieved by using Lagrange-Multipliers. For an unknown data point, equation 11 can be derived, using the Lagrange-multipliers α_i, again denoting values below ϵ with an asterisk. Those multipliers are nonzero for the so-called Support Vectors – values that determine the orientation of the regression hyper-plane.

$$\hat{y}_\mathrm{j} = \sum_{\mathrm{i}=1}^{N}(\alpha_\mathrm{i} - \alpha_\mathrm{i}^*)K(x_\mathrm{i}, x_\mathrm{j}) + b \tag{11}$$

K stands for the kernel function, representing an inner product for a linear model $K = x_\mathrm{i}^T x_\mathrm{j}$, but allowing also other functions to be introduced. This technique is called kernel trick and suits data that cannot be represented in a linear way by expanding the feature space (feature engineering). It holds the advantage that no manual feature extraction or data pre-processing is

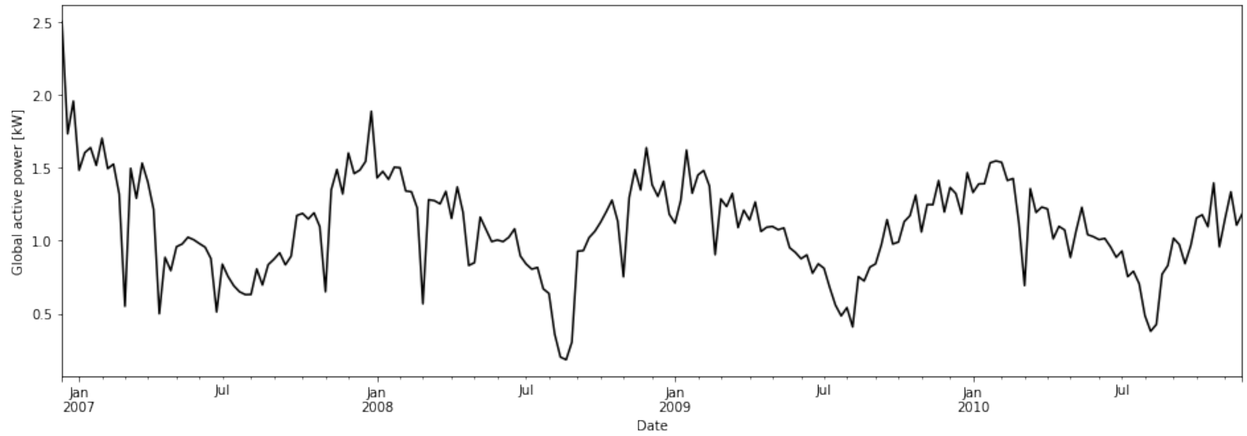

Figure 1: Complete load time series. Downsampled to weekly frequency for display purposes.

required and finding an appropriate kernel function K can be treated just as optimizing other hyper-parameters. The linear regression model can then be applied in the new feature space for finding a solution.

Other kernels than the linear kernel may thus also be feasible for mapping into an appropriate feature space, especially for non-linear relations. A widely applied and popular one is the Gauß-Kernel or Radial Basis Function (RBF) Kernel $K = \exp(-\frac{||x_\mathrm{i}-x_\mathrm{j}||^2}{2\sigma^2})$. Also the polynomial kernel $K = (1 + x_\mathrm{i}^T x_\mathrm{k})^d$ is often applied. Both approaches again introduce hyper-parameters that need to be tuned in the model selection process.

4 Data Exploration

The dataset consists of a multivariate time series for an individual house in the vicinity of Paris (Sceaux) and ranges from December 2006 to November 2010 (47 Months). With a sampling rate of 1 min it comprises 2,075,259 records and 7 features which label global active power, global reactive power, voltage, intensity and 3 subcircuits. Included in the load is the consumption of kitchen equipment, laundry facilities as well as climate, light and water heating, among others (Hebrail and Berard 2012). This study intends to only focus on the global active power as the aggregation of the complete power consumption. The dataset comprises missing values with a share of about 1.25 %, that need to be replaced by an interpolation method. It does not cover information about occupancy and behavioral schemes as well as weather conditions. That information is, however, included indirectly in the past observations of the time series.

This study focuses on a weekly periodicity, that can display annual consumption patterns and is less computationally expensive at the same time. Each weekly consumption value is calculated from the mean of its corresponding time steps. The entire time series is given in figure 1. The chart clearly shows a seasonal pattern with a period length of 52 to 53 weeks or respectively one year.

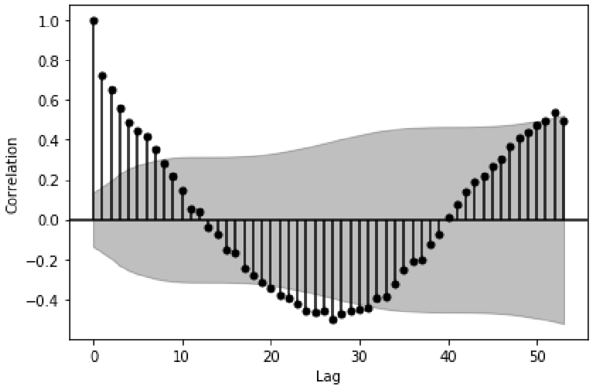

Figure 2: Autocorrelation function (ACF) and 95 % confidence interval.

The energy consumption is influenced by weather aspects, such as time of year and probably also by occupancy patterns, such as holidays and lifestyle. It clearly shows peaks in the win-

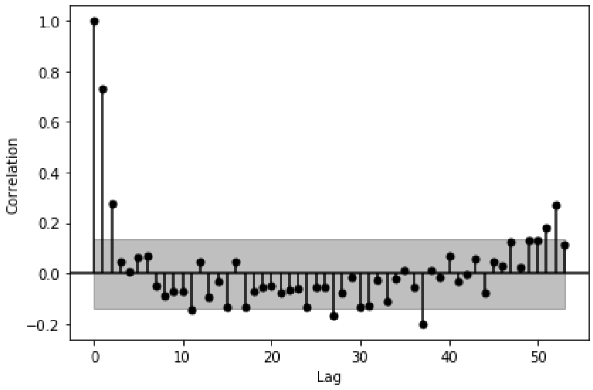

Figure 3: Partial autocorrelation function (PACF) and 95 % confidence interval.

ter months and minima in summer consistently over the years. Due to the seasonal behavior, a strong autocorrelation can be observed in figure 2 with a period of 52 to 53 weeks and countercyclical minima. Despite some significant values in between, a partial autocorrelating seasonal structure is present in figure 3 too. The partial autocorrelation function (PACF) is similarly to the autocorrelation function (ACF) a measure to evaluate the linear dependency of lagged variables, without considering the influence of values other than the directly observed ones though. I shows in this case, that values at a lag of one year contribute with additional information, influencing the value at the 0^{th} lag as well.

The seasonal decomposition in figure 4 underlines the seasonality considerations, by showing a clear seasonal component, approximately no trend and mostly random residuals. Interesting in this observation is the relatively low trend, compared to the deviations in the seasonal or residual component. It shows that the time series observed in this paper mainly consists of a dominant seasonal component and mostly randomly distributed residuals. The Ljung-Box-Test, applied to the seasonally differentiated time series, cannot reject the null-hypothesis of containing independent values and thus indicates that besides the seasonal component non-random information is only scarcely included in the IHEPC data. While modeling a seasonal component is fairly simple, white noise residuals are impossible to reproduce exactly. As shown later in this work, appropriate models are expected to mainly represent the seasonal component, but failing on the random noise component. Thus also non-complex models might work well with this time series structure when focusing on the seasonal patterns of energy consumption.

The stationarity of the time series is underlined by performing the Dickey-Fuller-Test, resulting in no unit root to be found. The time series can thus be considered stationary and ARIMA-models are applicable, perhaps using seasonal differencing.

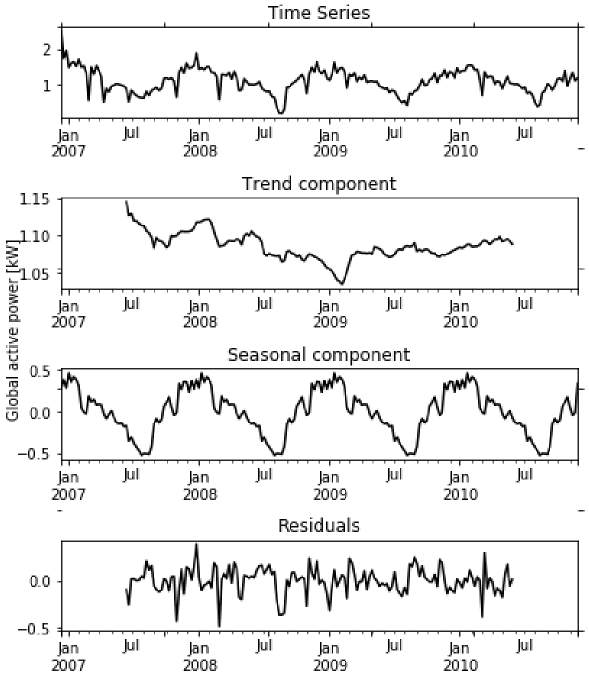

Figure 4: Seasonal decomposition of a weekly periodic load function for a yearly pattern.

Figure 5 shows the power consumption distribution for different temporal resolutions and groupings by displaying the median consumption and the quartiles. This helps to extract the consumption patterns in the household for different years, months, weekdays and hours. Beware that values for 2006 might be misleading, as they only cover the month December, which is a quite energy intensive time. Apart from that anomaly, the annual consumption is quite constant, indicating that no major changes were made to the energy facilities. Looking at the monthly distribution, the seasonal pattern can

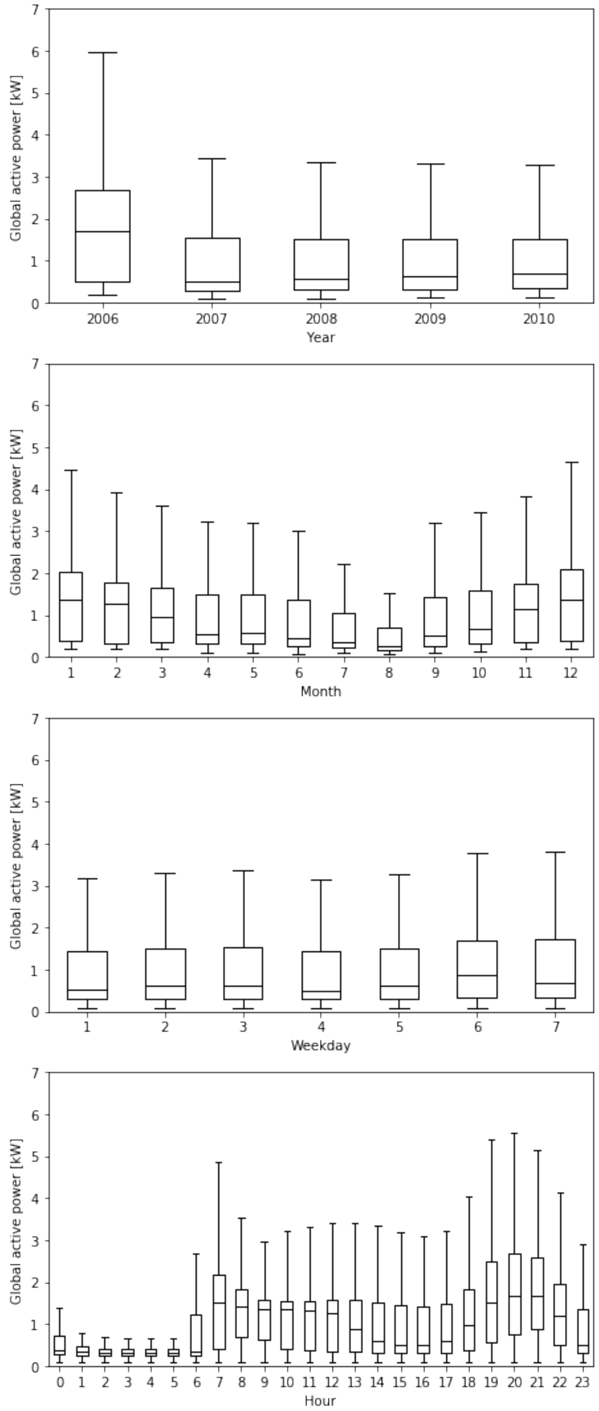

Figure 5: Load profiles for different temporal features.

be observed, that was discussed above in the time series decomposition with the lowest value and variance in August and maximum consumption in the winter months, which mainly might be caused by heating and electric light. The daily load values within a week are relatively similar with Saturday being slightly higher than the rest of the days. Hence, a small seasonal structure can be observed here as well. The hourly distribution shows the consumption pattern over a day. It includes a peak in the morning, followed by a plateau until noon, a slightly lower period then and finally another peak in the evening. At night, load and variance is low and displays the ground load being consumed by the electrical devices in the household. Variance is highest in morning and evening times, probably resulting from different behavioral patterns on weekdays and holidays.

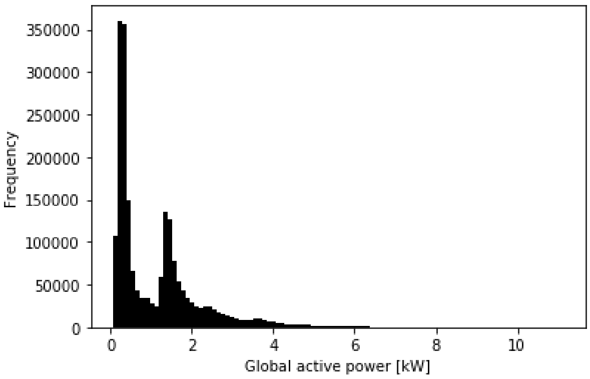

Figure 6: Distribution of load values.

The distribution of loads is also displayed in figure 6. Instead of being normally distributed, the histogram clearly shows two distinct peaks, where values occur more often apparently. This shape can be reproduced for any data frequency up to an hourly granularity, changing to Gaussian for lower frequencies. It thus indicates a daily consumption pattern probably with two prevailing modes of energy consumption. One might suggest that those could be a mode of absence (or night) and a mode of presence, which would occur at daytime. This behavior can be observed for different times of the day.

Those considerations make clear, that two major seasonalities are involved in the consumption: an annual and a daily. The first probably depends primarily on weather conditions, while the latter might be more due to behavioral characteristics and day-to-day routines. In contrast to weather data, which is easily quantifiable, occupancy data is way harder to obtain and unless it is not measured by smart home devices, not deployable in a practical context yet. However it is arguable, that both patterns are immanent in the measured consumption data and thus learnable by the algorithm to a certain extent without the provision of additional time series data.

In this study, only a weekly resolution is object of research, containing just the monthly seasonal pattern. In the case of higher granularity it is important to consider weekly and daily structures as well. This might influence the choice of the model, as multiple seasonalities are difficult to implement in an ordinary (seasonal) ARIMA model.

5 Methodology

Chujai et al. 2013 acts as a reference paper for this work, whose results are intended to be replicated. Besides ARIMA, it also investigates the applicability of ARMA models to the same dataset, which is, however, not part of this research. While the paper covers different frequencies of the time series, only the weekly data are considered in this work – mainly due to a limit of computational resources. It should be possible to apply the model used in the reference paper with only little differences, as the ARIMA configuration is given for weekly frequencies in the paper. Nevertheless, there might be variations as some processing steps are realized differently in Chujai et al. 2013 or not specifically stated at all.

The data pre-processing is carried out similarly to the reference paper, including resampling to the desired frequency and filling missing values with the last previous value. However while Chujai et al. 2013 first conduct the filling and then the resampling, the current research proposes it the other way around. This might lead to higher quality data, as filling the missing values – which is a quite inaccurate process – oftentimes becomes obsolete due to the data resampling, which calculates the mean value for each month anyways. Only in cases of complete months missing, filling the gaps is still necessary.

Another critical point in approaching the data is the period length of the weekly frequency. Unfortunately, a year cannot be divided into an integer of weeks. The accurate length is 52.18 weeks per year, however the considered seasonal differencing approaches and thus also ARIMA expect an integer period length. This might lead to inaccuracies as the actual seasonality and the one considered in the model shift over time. It would be intuitive to choose 52 as an integer value for period length, however Chujai et al. 2013 chose 53 in their work for unknown reason. In order to reproduce the results of the ARIMA model, 53 is used in this paper as well. However, when applying the SVR model, 52 is used instead.

For model evaluation, the dataset is split into two sections, the training set on which the model is trained, and the test set, on which the model performance is evaluated. As it is crucial to maintain the temporal structure in a time series, it is necessary that the test set follows the training set in contrast to other Machine Learning problems. The forecast itself is done with a rolling window, starting at the beginning time of the test series and iterating forward using the previously predicted values (not the values actually included in the test set).

The split is done on 1^{th} January 2010, leaving 11 Months for testing and prior 36 Months for training. In order to be able to compare the results, this split is adopted in the current study. However it arguably has advantages to test on multiple test sets (e.g. through forward chaining in a rolling window approach) to obtain a more holistic measure of model applicability and not only considering the goodness within one time span. This is not carried out in this research.

The quality of the model and the goodness of the fit regarding the test set is evaluated with the root mean squared error, which is calculated as in equation 12.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(y_\text{i} - \hat{y}_\text{i})^2}{N}} \tag{12}$$

N is the sample size, $\hat{y}$ the predicted value and y the actual value. As Chujai et al. 2013 use the same evaluation metric, it is possible to compare results.

The different models are calibrated on the training dataset using different approaches, as outlined in the following sections.

ARIMA Model Selection based on the AIC

As noted earlier, Brockwell and Davis 2016 propose the use of the AIC in order to determine appropriate models. As this approach is used in the reference paper, it is applied here as well. Performing a Grid Search, different ARIMA models with different parameter combinations are deployed on the training dataset while measuring the performance with the AIC. The combination resulting in the lowest AIC is considered as the best model and then tested on the out of sample values.

ARIMA Model Selection based on ACF and PACF Functions

Manual model determination is possible as well, although this method is not suitable for a completely automated approach. The manual evaluation relies on the time series itself as well as its' ACF and PACF plots and reads the values for p and q from there. First, any seasonal component is eliminated by differencing the time series, which sets the hyperparameter d and achieves stationarity in the data. After that, the ACF determines the MA part with parameter q. Figure 10 shows the auto-correlation and its' significance level. An MA(q) model now requires the lags to be insignificant before the q$^{\text{th}}$ order, containing no significant lags prior to this point. Similarly, the AR(p) part results from the PACF in figure 11, containing no significant lags prior to the p$^{\text{th}}$ lag.

In this study, this method is used for the validation of the automatically derived models.

SVR Model Selection

While the ARIMA models are derived by consulting the AIC in the grid search, the Support Vector Regression model is determined by cross-validation. Within the training dataset, an expanding window approach is used to create multiple series sets and evaluate them on a validation set with the same length as the final forecasting period. This enables a finer measurement of the validation error, as it is averaged over all expanding windows.

SVR requires a set of parameters to be determined: the regularization parameter C, the error range ϵ, the rolling window size N and finally also the choice of the kernel function K. By measuring the RMSE (Equation 12) for each expanded window and averaging all values, a total RMSE is calculated which reflects the goodness of the fit. The parameter combination resulting in the lowest RMSE is considered as the best model and applied to the test dataset.

The forecast itself is done by using rolling windows, similarly to the earlier discussed ARIMA method, thus predicting future values recursively by their preceding data. For the final accuracy measurement, those forecasts are compared with the same test dataset as used in Chujai et al. 2013 for the ARIMA evaluation. In contrast to ARIMA, SVR requires scaled values, so that a backward transformation is necessary in order to compare correctly.

6 Experimental Evaluation

ARIMA Models

Model	AIC	RMSE
ARIMA$(1,0,1)(0,0,0)_{53}$	-16.7	0.3263
ARIMA$(1,0,1)(0,1,0)_{53}$	40.7	0.1656
ARIMA$(0,0,0)(0,1,0)_{53}$	42.7	0.1656

Table 1: Examined ARIMA models and their respective AIC and RMSE values.

In total, 3 different ARIMA settings were tested on the time series. Their configurations and their respective accuracy metrics are illustrated in table 1. Strikingly, all models regardless of their derivation method, consider only a very small amount of lags for the AR and MA processes.

The first model – $\mathrm{ARIMA}(1,0,1)(0,0,0)_{53}$ – was derived by grid search and holds with $\mathrm{AIC} = -16.7$ the lowest AIC. According to Brockwell and Davis 2016, this should be the model that fits the time series best. However considering the rather high RMSE of 0.3263 in comparison to the other performances, it seems to miss some essential relations. Looking at the forecast in figure 7, showing on the left side the training series and on the right the test series, it is clear that this configuration is not able to predict the time series adequately. The reason for this lies in the missing seasonal differentiation in the ARIMA model, which is a clear contradiction to the correlations observed in the ACF/PACF plots (Figures 2 and 3). With just a lag of one for AR and MA, to a certain extent the prediction just follows the previously observed values of the time series. This trivial model might work well in the in-sample dataset, where the actual observations are given, but performs poorly on the out of sample data. It is thus not surprising, that the accumulated RMSE in figure 12 is low at the beginning, but begins to rise after some weeks and performs worse for longer forecast horizons, due to the short term dependencies of the model. This model is just applicable for a low forecasting range and misses the seasonal pattern completely.

The results of this model lead to the conclusion, that the AIC as an evaluation metric has some constraints when it comes to seasonal patterns. A simple grid search might not be sufficient for finding the best model, as it allows the fitted series to closely follow the training data, resulting in a low AIC for a model, that does not cover all essential relations in the dataset sufficiently. Worth considering might be an expanding window approach with a larger forecast horizon in order to consider the seasonality properly.

The second model tested is $\mathrm{ARIMA}(1,0,1)(0,1,0)_{53}$, proposed by Chujai et al. 2013 and producing an AIC of 40.7. Differently to the above mentioned configuration, here, a seasonal differentiation is applied beforehand. Although the AIC is larger it results in a smaller RMSE of 0.1656 for the forecasting period, thus performing better on the test data. The prediction can be seen in figure 8.

It can be seen clearly, that the seasonal pattern is adopted by the model and leads to a far better representation of the ongoing periodical series, also long-term. Figure 12 shows its performance compared to other tested models for different forecast horizons.

The third model was derived by manual observation of the time series and its' ACF and PACF plots. To eliminate seasonal patterns, as seasonal difference was applied to the series, resulting in the values displayed in figure 9 and in the seasonal parameter $D = 1$. On the basis of this de-seasonalized series, the model parameters were derived from the ACF and PACF shown in figures 10 and 11. As neither in ACF nor in PACF significant correlations occurred to the lags directly before, a model configuration of $\mathrm{ARIMA}(0,0,0)(0,1,0)_{53}$ was deduced. It can be seen as a seasonal random walk process, just predicting the values of the same times in the previous year and deviating by a white noise residual.

Although this model is fairly trivial, it produces an AIC of 42.7 and an RMSE of 0.1656, equal to the $\mathrm{ARIMA}(1,0,1)(0,1,0)_{53}$ process. Indeed, the forecast is very similar to the one proposed by Chujai et al. 2013 and no identifiable differences can be seen to the forecast in figure 8 and to the error development in figure 12. However due to its' setup itself, this model is not capable to explain any deviation from the seasonal pattern at all. Most of them stay random and unpredictable.

What the results of the three ARIMA models show, is what was indicated already above in the Data Exploration regarding the character of the time series: In the case of the IHEPC data, seasonal patterns play a dominant role in the energy demand of the household, causing the

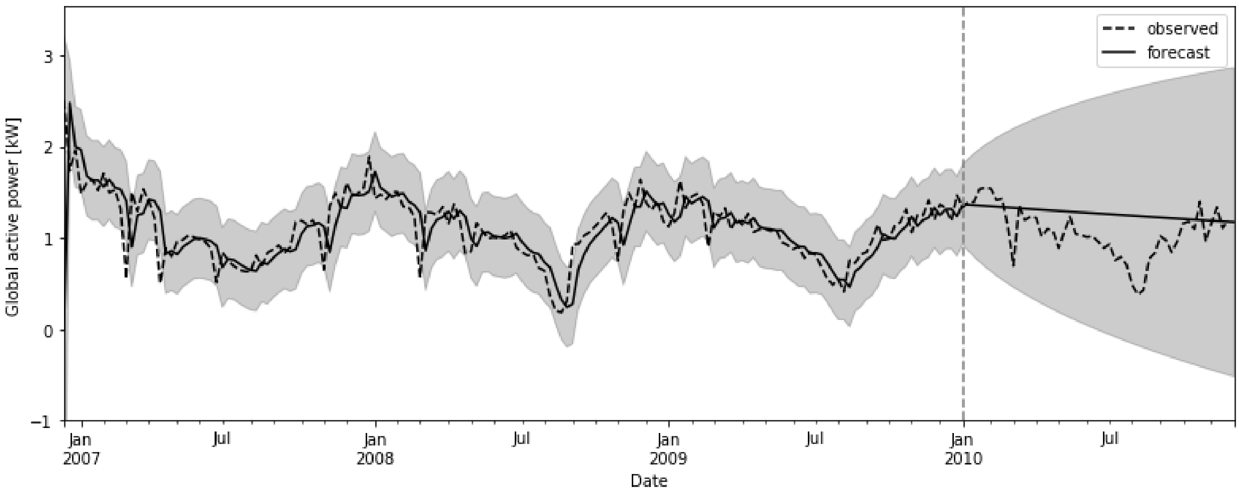

Figure 7: Fitted curve and prediction with ARIMA(1,0,1) proposed by the AIC evaluation.

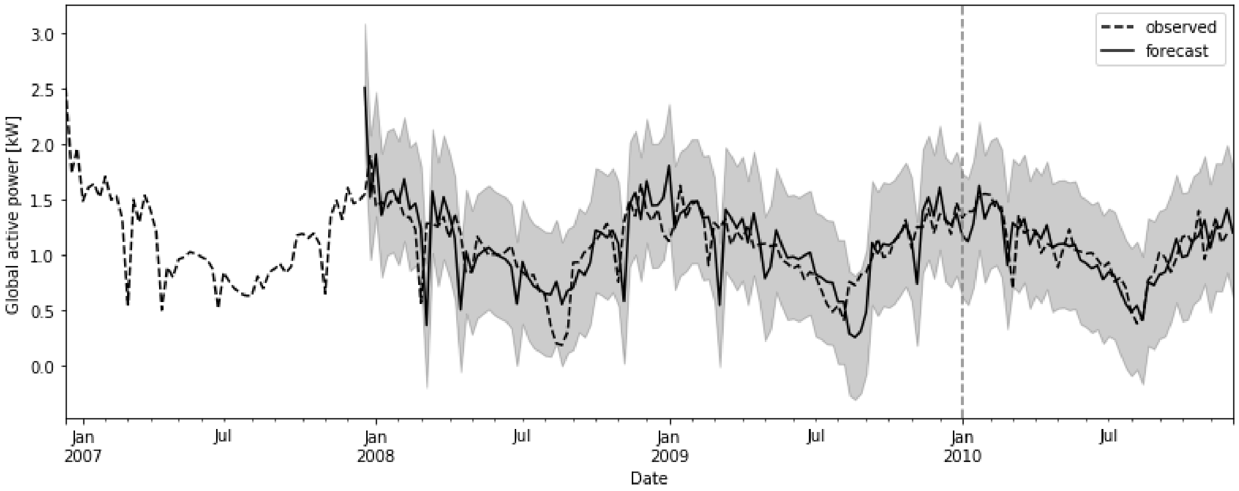

Figure 8: Fitted curve and prediction with ARIMA(1,0,1)(0,1,0)53 proposed by Chujai et al. 2013.

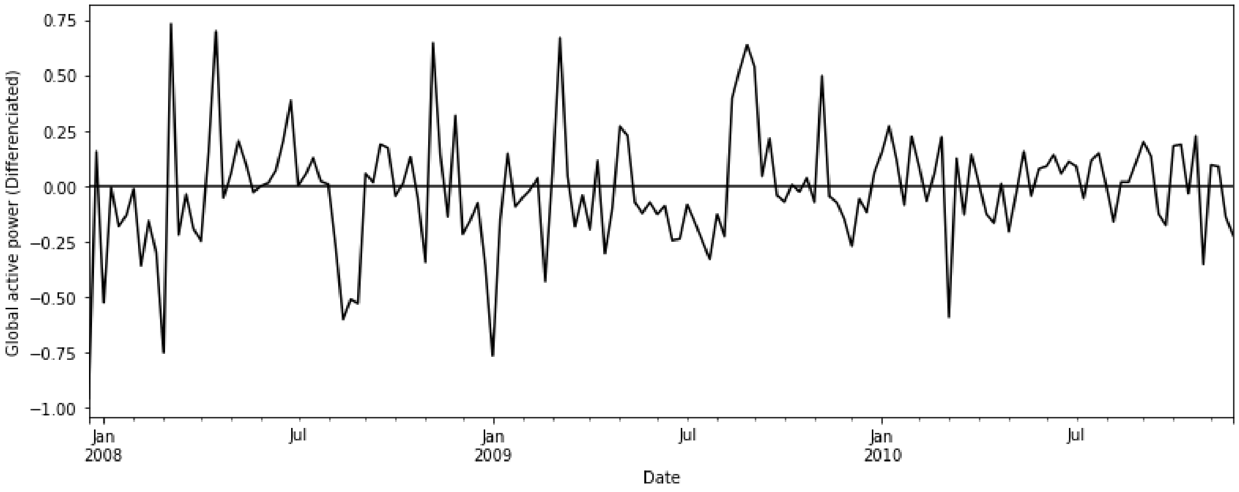

Figure 9: Seasonally differentiated time series.

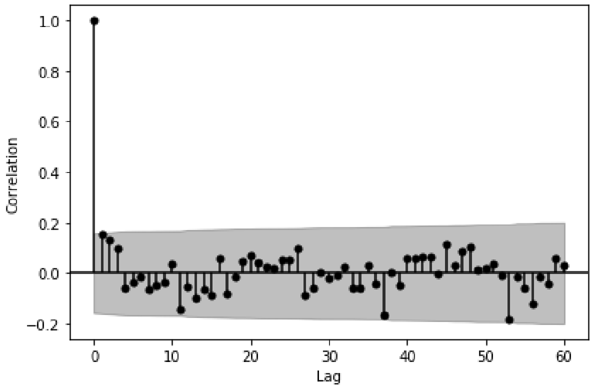

Figure 10: Autocorrelation function for seasonally differentiated time series.

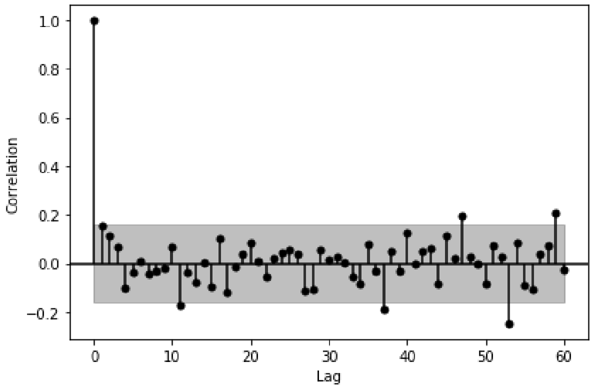

Figure 11: Partial autocorrelation function for seasonally differentiated time series.

residuals to be mostly randomly distributed after all. This causes statistical models to model just the seasonal patterns with quite high accuracy, being unable to find relations within the non-seasonal data though. Arguably this is due to the low resolution of the time series used in this paper, but also to the large impact seasonal weather conditions might have on energy consumption. The seasonal random walk might thus be a simple, yet powerful model making more complex approaches for this resolution and forecast horizon obsolete.

The performance of all models is displayed in figure 12, where the accumulated RMSE values are shown for different forecasting horizons, starting at the time of the split. The error development of ARIMA$(0,0,0)(0,1,0)_{53}$ and ARIMA$(1,0,1)(0,1,0)_{53}$ is so similar, that it is not distinguishable in the graph. As can be seen, ARIMA models with $D = 1$ outperform those with $D = 0$ in the long run clearly, but ARIMA$(1,0,1)(0,0,0)_{53}$ shows better results for the first weeks of the forecast. If that is just due to the choice of this specific date of the split or a general phenomenon is not clear from those results. However it seems obvious that models with a seasonal pattern perform long-term better than those without.

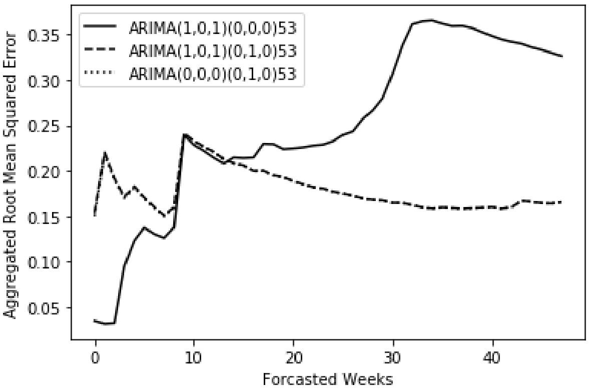

Figure 12: Aggregated RMSE values for different forecasting periods.

SVR Models

Model	RMSE
(A) RBF, $C = 0.1$, $\epsilon = 0.01$, $N = 35$	0,1683
(B) RBF, $C = 1$, $\epsilon = 0.01$, $N = 10$	0,2061
(C) RBF, $C = 0.1$, $\epsilon = 0.01$, $N = 35$ (seasonal diff.)	0,1927

Table 2: Examined SVR models and their respective RMSE performance.

For comparison with ARIMA, three different SVR models were tested on the same dataset: an ordinary SVR (A), an SVR with seasonal differencing beforehand (B) and model A applied on a seasonally differentiated time series (C). The performance of all three models and their choice of parameters is outlined in table 2. All models were trained on the same training dataset used for training ARIMA but using grid search and expanding windows to obtain a more accurate evaluation measure (RMSE).

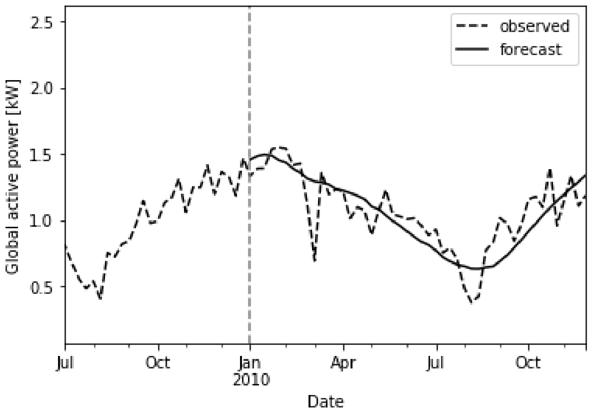

Figure 13: Prediction with Gaussian Kernel, $C = 0.1$, $\epsilon = 0.01$, window size 35 (A).

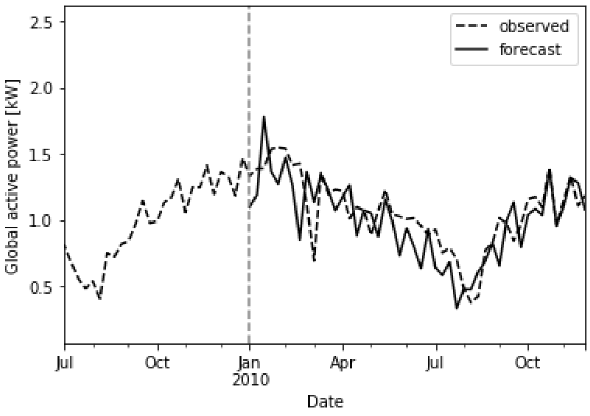

Figure 14: Prediction with Gaussian Kernel, $C = 1$, $\epsilon = 0.01$, window size 10, seasonal differentiated (B).

Model A used the parameters $C = 0.1$, $\epsilon = 0.01$ and $N = 35$ while choosing the Radial Basis Function for the kernel in the grid search. The resulting forecast is depicted in figure 13. It shows a quite smooth curve that covers the seasonal pattern but does not follow the small scale twists of the load. With an RMSE of 0.1683, it still produces an accuracy that is comparable to the performance of the ARIMA models. Interestingly it does not require seasonal differencing before and still uses an optimal window size of approximately half a year.

Model B was derived from grid search as well, however now performed on seasonal differentiated data. This step resembles the use of $D \geq 1$ in ARIMA and should test if the model can replicate the non-seasonal patterns of the time series. The optimal model proposed by grid search holds the parameters $C = 1$, $\epsilon = 0.01$, $N = 10$ and the Radial Basis Function as well. With an RMSE of 0.2061 it performs worse than without seasonal differencing however and is not able to represent the small deviations either (see figure 14).

It is not completely clear why the performance is worse, but it could be caused by the small training dataset that was used to derive the hyperparameters. Due to the seasonal differencing, there was a whole year less available to train the model, resulting in less expanding windows to calculate the evaluation criterion RMSE and thus the optimal set of parameters. It does not implicate that the seasonal difference is an inadequate operation in this case per se.

Lastly it was examined how the differencing step influences the model performance after all. Model A was thus not given the original time series but a differentiated instead. The forecast produces an RMSE of 0.1927 and can be examined in figure 15. Clearly, the performance is worse than for model A, however slightly better then model C. It is obvious that the model was not specifically tuned for the differentiated time series, as that would be the setup of model B with all its' implications.

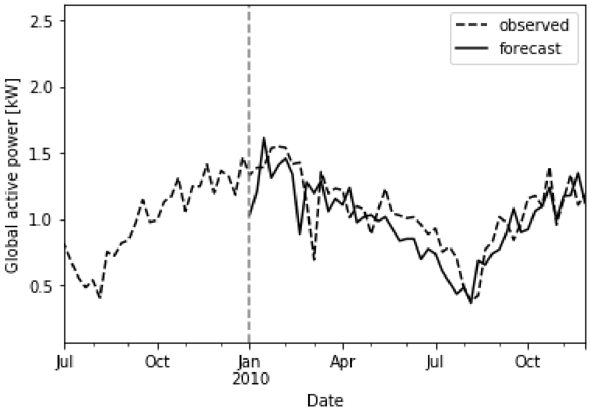

Figure 15: Prediction with Gaussian Kernel, $C = 0.1$, $\epsilon = 0.01$, window size 35, seasonal differentiated (C).

Comparing the accumulated errors in figure 16, it can be seen clearly that the SVR model applied on the ordinary time series (A) outperforms B and C for all time steps within the forecasting horizon.

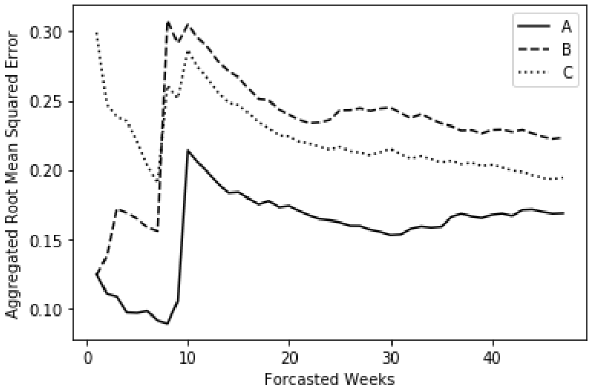

Figure 16: Comparison of error developments.

7 Conclusion

The presented work outlined the applicability of ARIMA and SVR models with different configurations respectively for the forecast of a household's energy consumption. The character of the uni-variate time series was dominated by its' seasonal component, while deviations from this patterns were small and mainly randomly distributed. Those residuals were difficult to predict, as they may result from behavioral patterns from the inhabitants amongst other influences.

Hence, ARIMA models with $D = 1$ performed best among other configurations, as they were capable of modeling the seasonal component of the load while non-seasonal values are mainly ignored. The obtained models are fairly trivial as they mainly predicted the test data using values from the previous year, yet obtaining high accuracies in comparison with other parameter configurations. SVR models predicted primarily the seasonal pattern as well and resulted in similar accuracies as the ARIMA models.

This work shows, that even trivial models are capable of forecasting the energy load to a certain extend. The seasonal random walk seems to be a good representation of the current time series with a weekly frequency. However it is arguable if the derived models perform in the same manner for different frequencies and other forecasting horizons.

In comparison with other resources on this dataset that consider a weekly frequency, the present models perform similarly. Le et al. 2019 conducted research on Linear Regression, LSTM, CNN-LSTM and EECP-CBL for a two year forecast on the IHEPC dataset with a multi-variate approach using all available features, obtaining only for the latter RMSE scores comparable to the results of ARIMA or SVR. However the forecast horizon is different to the present work, so there is a chance of variations. Kim and Cho 2019 followed a multi-variate approach too, using a CNN-LSTM on the same forecast horizon as this paper, obtaining better RMSE scores than the ones obtained in this work. Like in the present work, a uni-variate time series of the global active power was examined in Mocanu et al. 2016 with the same forecast horizon, using ANN, SVM, RNN, CRBM and FCRBM. ANN and RNN perform worse than the models presented in this paper, while the others produce a similar accuracy.

ARIMA as a fairly simple statistical model and SVR thus perform similarly or even better on the IHEPC dataset than complicated Deep Learning models. This might, however, change when other frequencies are considered. The IHEPC dataset is just one among many household load time series, and different buildings yield different climate conditions, occupancy patterns, technical facilities or socio-demographic characteristics. It would be interesting for followup research to investigate the impact of other frequencies and other test samples or even completely different time series. Probably changing those circumstances entails the need for other parameter configurations or even more sophisticated models.

8 References

Albert, Reinhard et al. (2016). *Klimaschutzplan 2050 der Bundesregierung - Diskussionsbeitrag des Umweltbundesamtes*. Tech. rep. Umweltbundesamt, Fachgebiet I 2.2.

Beliaeva, Nataliia, Anton Petrochenkov, and Korinna Bade (2013). "Data Set Analysis of Electric Power Consumption." In: *European Researcher* 61, pp. 2482–2487. DOI: 10.13187/er.2013.61.2482.

Box, George E. P., Gwilym M. Jenkins, and Gregory C. Reinsel (2008). *Time series analysis: forecasting and control*. Fourth edition. Wiley series in probability and statistics. Hoboken, New Jersey: Wiley, a John Wiley and Sons, Inc. ISBN: 9781118619063. DOI: http://dx.doi.org/10.1002/9781118619193.

Brockwell, Peter J. and Richard A. Davis (2016). *Introduction to Time Series and Forecasting*. Third edition. Springer Texts in Statistics. Springer. ISBN: 978-3-319-29854-2. DOI: http://dx.doi.org/10.1007/978-3-319-29854-2.

Chujai, Pasapitch, Nittaya Kerdprasop, and Kittisak Kerdprasop (2013). "Time series analysis of household electric consumption with ARIMA and ARMA models." In: *Lecture Notes in Engineering and Computer Science* 2202, pp. 295–300.

Commandeur, Jacques J. F. and Siem Jan Koopman (2007). *An Introduction to State Space Time Series Analysis*. Practical Econometrics Series. OUP Oxford. ISBN: 9780199228874.

Cristianini, Nello and John Shawe-Taylor (2000). *An introduction to support vector machines : and other kernel-based learning methods*. Cambridge: Cambridge University Press. ISBN: 9780511801389.

Deb, Chirag, Fan Zhang, Junjing Yang, Siew Lee, and Shah Kwok Wei (2017). "A review on time series forecastingtechniques for building energy consumption." In: *Renewable and Sustainable Energy Reviews* 74. DOI: 10.1016/j.rser.2017.02.085.

Hebrail, Georges and Alice Berard (2012). *Individual household electric power consumption dataset*. Website. Retrieved from https://archive.ics.uci.edu/ml/datasets/Individual+household+electric+power+consumption (04.05.2020). Irvine, CA: University of California, School of Information and Computer Science.

Khan, Zulfiqar, Tanveer Hussain, Amin Ullah, Seungmin Rho, Miyoung Lee, and Sung Baik (2020). "Towards Efficient Electricity Forecasting in Residential and Commercial Buildings: A Novel Hybrid CNN with a LSTM-AE based Framework." In: *Sensors* 20, p. 1399. DOI: 10.3390/s20051399.

Kim, Tae-Young and Sung-Bae Cho (2019). "Predicting residential energy consumption using CNN-LSTM neural networks." In: *Energy* 182, pp. 72 –81. ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2019.05.230. URL: http://www.sciencedirect.com/science/article/pii/S0360544219311223.

Le, Tuong, Minh Vo, Bay Vo, Eenjun Hwang, Seungmin Rho, and Sung Baik (2019). "Improving Electric Energy Consumption Prediction Using CNN and Bi-LSTM." In: *Applied Sciences* 9, p. 4237. DOI: 10.3390/app9204237.

Makridakis, Spyros and Michèle Hibon (1997). "ARMA Models and the Box–Jenkins Methodology." In: *Journal of Forecasting* 16.3, pp. 147–163. DOI: 10.1002/(SICI)1099-131X(199705)16:3<147::AID-FOR652>3.0.CO;2-X.

Marino, Daniel, Kasun Amarasinghe, and Milos Manic (2016). "Building Energy Load Forecasting using Deep Neural Networks." In: DOI: 10.1109/IECON.2016.7793413.

Mocanu, Elena, Phuong Nguyen, Madeleine Gibescu, and Wil Kling (2016). "Deep Learning For Estimating Building Energy Consumption." In: *Sustainable Energy, Grids and Networks* 00, pp. 1–10. DOI: 10.1016/j.segan.2016.02.005.

Nejat, Payam, Fatemeh Jomehzadeh, Mohammad Mahdi Taheri, Mohammad Gohari, and Muhd Zaimi [Abd. Majid] (2015). "A global review of energy consumption, CO2 emissions and policy in the residential sector (with an overview of the top ten CO2 emitting countries)." In: *Renewable and Sustainable Energy Reviews* 43, pp. 843 –862. ISSN: 1364-0321.

DOI: https://doi.org/10.1016/j.rser.2014.11.066.

Puspita, Verlly and Ermatita Ermatita (2019). "Time Series Forecasting for Electricity Consumption using Kernel Principal Component Analysis (kPCA) and Support Vector Machine (SVM)." In: *Journal of Physics: Conference Series* 1196, p. 012073. DOI: 10.1088/1742-6596/1196/1/012073.