SEVENTH FRAMEWORK PROGRAMME

MIRACLenergy

Balancing energy supply and demand

MIRACLE

Micro-Request-Based Aggregation, Forecasting and Scheduling of Energy Demand, Supply and Distribution

Specific Targeted Research Project: 248195

D4.1: State-of-the-Art Report on Forecasting

Work Package 4: Forecasting

Leading partner: TUD

25th June, 2010

Version 1.1

DOCUMENT INFORMATION	
ID	D4.1: State-of-the-art report on forecasting
Work Package(s)	WP4: Forecasting
Type	R
Dissemination	PU
Version	1.1
Date	25th June 2010
Author(s)	Lars Dannecker (SAP), Matthias Boehm (TUD), Ulrike Fischer (TUD), Frank Rosenthal (TUD), Gregor Hackenbroich (SAP), Wolfgang Lehner (TUD)
Reviewer(s)	Bogdan Filipic (JSI)

“The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The above referenced consortium members shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law.

A Survey of Forecast Models for Energy Demand and Supply

Lars Dannecker[2], Matthias Boehm[1], Ulrike Fischer[1], Frank Rosenthal[1],
Gregor Hackenbroich[2], and Wolfgang Lehner[1]

[1] TU Dresden, Database Technology Group
[2] SAP Research CEC Dresden

Abstract

In the energy domain, one of the most important goals is the integration of more renewable energy sources (RES, e.g., windmills, solar panels). Unfortunately, most RES depend on external factors such as wind speed and the amount of sunlight. Hence, available power from RES cannot be planned as traditional energy sources and thus, there is the need of balancing energy demand and supply. We address this requirement within the MIRACLE (Micro-Request-Based Aggregation, Forecasting and Scheduling of Energy Demand, Supply and Distribution) project with a micro-request-based approach, where acceptable flexibilities (e.g., timeshifts or variable amount) can be specified by consumers and producers along with concrete energy demand and supply, respectively. These flexibilities allow for fine-grained scheduling and thus, balancing of demand and supply. Accurate and efficient forecasts for short-term and long-term horizons of energy consumption and production as well as for requests with timeshifts are a fundamental precondition for dynamic and fine-grained scheduling of energy demand and supply.

In this survey, we give a detailed overview of forecast models for energy demand and supply. First, we reveal typical data characteristics of energy demand and supply. Second, we describe the mathematical background of time series forecasting in general. Furthermore, we review existing domain-specific techniques of forecasting energy demand, supply and prices as well as how these techniques can be integrated into a system architecture of a data management system. Third, we select representative forecast models from the main categories of existing techniques and evaluate their accuracy with regard to different time horizons. Fourth, we identify major challenges and open problems that should be addressed in order to enable accurate and efficient forecasting as a fundamental prerequisite for scheduling energy demand and supply. Finally, the scheduling of energy demand and supply will allow (1) to smoothen cost-extensive peaks, (2) to integrate more renewable energy sources, and (3) to balance energy demand and supply.

Contents

1 Introduction **1**

2 Energy Data Analysis **3**
2.1 Data Set Descriptions 3
2.2 Data Characteristics 4

3 Time Series Forecasting **9**
3.1 Autoregressive Models 11
3.2 Moving Average and Exponential Smoothing 16
3.3 Machine Learning Techniques 20
3.4 Methods for Parameter Estimation 27
3.5 Measuring the Accuracy 31
3.6 Summary and Discussion 35

4 Related Work in Energy Forecasting **37**
4.1 Forecasting Energy Demand 37
4.2 Forecasting Energy Supply 47
4.3 Forecasting Energy Prices 48
4.4 System Architecture 50
4.5 Summary and Discussion 53

5 Experimental Evaluation **54**
5.1 Experimental Setting 54
5.2 Forecasting with TESM 56
5.3 Forecasting with HWT 57
5.4 Forecasting with EGRV 58
5.5 Comparison of Forecast Models 60

6 Characteristics and Challenges **62**
6.1 Characteristics 62
6.2 Research Challenges 63

7 Conclusions **65**

1 Introduction

The energy market is changing from a static (one-day-ahead) market with fixed actors to a flexible and highly dynamic (real-time) marketplace, where the consumers and producers are free to choose the energy supplier in a fine-grained manner. The reasons for this liberalization are twofold. On the one side, technical preconditions such as current smart meter technology at the household level enable a more dynamic demand and supply negotiation. On the other side, regulatory policies force this market change. For example, there are European policies that focus on creating a competitive internal energy market that enables dynamic trading. In addition, the general goal of integrating more renewable energy sources exists in order to reduce the dependence on traditional energy sources. Unfortunately, most renewable energy sources (RES; e.g. windmills, solar panels) pose the challenge that the production depends on external factors such as wind speed and the amount of sunlight. Hence, available power from RES cannot be planned as traditional energy sources. As a result, there is the need for more fine-grained management of energy demand and supply as well as a corresponding balancing at the level of energy brokers.

We address this requirement of real-time balancing of energy demand and supply within the MIRACLE (Micro-Request-Based Aggregation, Forecasting and Scheduling of Energy Demand, Supply and Distribution) project. Our main goal is to develop a conceptual and infrastructural approach that allows energy distribution companies to efficiently manage higher amounts of renewable energy and balance supply and demand. The core idea of this project is the micro-request-based approach, where acceptable flexibilities (e.g., timeshifts or variable amount) can be specified by consumers and producers along with concrete energy demand and supply, respectively. These flexibilities allow for fine-grained scheduling and thus, balancing of demand and supply at the level of energy brokers. As a result, we are able (1) to smoothen cost-extensive peaks of energy demand that currently can only be satisfied by traditional energy sources, (2) to integrate more renewable energy sources, and (3) to balance energy demand and supply at lower costs for all involved actors and thus, to achieve an active customer involvement.

In this survey, we exclusively focus on the aspect of forecasting energy demand and supply because accurate forecasts for short-term and long-term horizons are a fundamental precondition for dynamic and fine-grained scheduling of energy demand and supply. Therefore, separated forecasts are required for energy consumption and production as well as for requests with timeshifts. Hence, accurate and efficient forecasting of energy demand and supply is a key enabling technology.

In general, there is plenty of existing work on time series analysis and forecasting. Based on the specific characteristics of energy demand and supply data combined with the need for very accurate forecasts, many forecast models and techniques, tailor-made for the energy domain, have been proposed as well. Unfortunately, there is no common sense on the selection of an appropriate forecast model for energy data. This makes it hard to reuse known results. In addition to this diversity of forecast models, we observe that, despite similar data characteristics, many proposed solutions are tailor-made for regional data (e.g., forecasting country-specific data) or specific time horizons (short-term, long-term). These assumptions might fail when facing an international, flexible and highly dynamic marketplace for energy demand and supply.

In consequence, the main research questions concerning energy forecasts are (1) what are the best performing forecast models for energy demand and supply with regard to accuracy, and (2) what challenges and open problems arise in the context of energy forecasting in a flexible and dynamic marketplace. In order to focus on these research questions and to address the mentioned problems, we make the following contributions that are also reflected in the structure of this survey:

- First of all, in Chapter 2, we reveal common data characteristics of energy demand and supply. There, we introduce three real-world, multi-regional data sets that are used throughout this survey.

- In Chapter 3, we describe the general mathematical background of time series forecasting. This includes a description of the general analysis process, a detailed classification of common forecast models as well as typically used error metrics.

- More specific, in Chapter 4, we survey existing work on forecasting of energy demand and supply as well as forecasting in related domains. Essentially, we classify typical forecast models for energy data and we identify representative models for our evaluation.

- Subsequently, we test the identified forecast models on the three introduced real-world data sets in Chapter 5. We evaluate the different forecast models using the introduced error metrics. We explicitly do not want to introduce a new forecast model but to evaluate existing techniques and conclude, which models are most appropriate for certain data characteristics and time horizons.

- In Chapter 6, we reveal the major challenges and open problems that exist in the area of forecasting energy demand and supply within a flexible and dynamic marketplace. Summarizing the characteristics of energy data as well as the balancing-process-requirements, we derive research challenges and interesting aspects of future work.

- Finally, we conclude the survey in Chapter 7 with a summary of our main results.

2 Energy Data Analysis

As background information for understanding the discussion on existing techniques of energy forecasting, in this chapter, we reveal important energy data characteristics using three data sets.

2.1 Data Set Descriptions

Essentially, we used three real-world energy production and consumption data sets in our energy data analysis. One of them is a private data source of a MIRACLE consortium partner, while the other two are publicly available. In the following, we describe these data sources in more detail.

Data Set A: National Grid Electricity Demand

The first data set was obtained from the National Grid organization. This data set includes metered electricity demand of the United Kingdom (UK) and is publicly available at [Nat10]. Essentially, it contains different demand indicators, where we used the INDO[1] measure and E&W[2] (historic measure, but similar to INDO) measure for periods where INDO was not available.

In detail, this data set contains two data granularities. First, there is historical total demand data from April 1st 1971 until December 31st 2009 in a per-day resolution. Second, it also includes more recent total demand data from April 1st 2001 until December 31st 2009 in a 30 min resolution. Due to the aggregation over the whole United Kingdom and its fine-grained resolution, it allows for very accurate analysis of total demand and enables conclusions on typical *temporal* time series characteristics.

Data Set B. CRES Test Site Supply and Demand Data

Our second data set was provided by the MIRACLE partner CRES (Center For Renewable Energy Sources) from Greece. This private data set contains very detailed metered demand and supply at the consumer and producer level, respectively. The data was obtained for a single building (offices and laboratories of CRES departments PV and DER) with 440 m^2 building area of one of our test sites.

In contrast to the first data set, this data source exhibits a much finer aggregation level and thus, allows for the analysis of other important issues. In detail, this data set

[1] "**INDO** - Published BMRA Initial Demand Outturn based on National Grid operational generation metering. EXCLUDES Station Load, Pump Storage Pumping and Interconnector Exports" [Nat10]

[2] "**E&W** - For legacy users this is a continuation of what would be Pre-BETTA INDO." [Nat10]

includes metered supply and demand from January 11th 2008 to December 16th 2008 in a 1 min resolution. It includes three indicators. First, the supply is represented by the production of a 22 kWp PV (photo-voltaic) panel, installed on the roof and the front side of the building (Ppv). Second, the heating and cooling demand are covered by one central heat pump (Pheatpump). Third, the rest of the demand was included as an additional indicator. In conclusion, this data set stands in contrast to the aggregated demand data because such a laboratory exhibits different typical characteristics (Ploads).

Data Set C. Worldwide Electricity Demand

The third data set was obtained from the US Energy Information Administration and is public available at [US 10]. This data set includes metered world-wide electricity consumption and generation as well as many other indicators such as capacity, imports, export or distribution losses. Beside electricity, this data set also contains information about petroleum, natural gas and coal.

In detail, this data set contains many analysis dimensions. First, data is available from 1980 until 2008 in an annual resolution. Second, it contains hierarchical regional information from world-wide over continental to individual countries. Third, it further distinguishes different categories of electricity sources such as renewables or nuclear. In conclusion, this data set allows for a detailed time series analysis of regional and categorical aspects, while it is unfortunately, pretty coarse-grained with an annual resolution.

2.2 Data Characteristics

In this section, we consolidate our main findings on characteristics on energy demand and supply data from the three data sets. This should enable a better understanding of the specific characteristics of energy demand and supply data that reasoned the development of tailor-made forecast models for this domain.

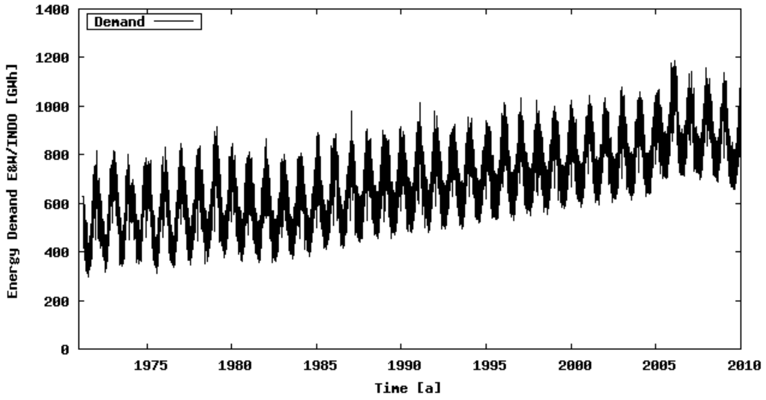


Figure 2.1: Data Set A – Linear Trend of Total Energy Demand UK

First of all, we use data set A in order to reveal temporal fluctuations. This includes the general trend as well as multiple seasonal aspects. Figure 2.1 illustrates the historic development of energy consumption. Therefore, we aggregated the total energy demand per day. Essentially, we observe a weak linear trend over the last 40 years. Furthermore, we see that within the last 5 years, this trend decreases such that the differences between days of subsequent years are almost neglectable for short- to mid-term forecast horizons.

Furthermore, there is an obvious additive seasonal component. In order to investigate this issue in more detail, Figure 2.2 illustrates the total energy demand for a shorter time window over the last eight years. Again, we aggregated the total energy demand per day. Now, we observe mainly two characteristics. First, there is an annual season in the form of a dependence on the weather. Thus, the energy consumption during winter days is significantly higher than the demand in summer days. Second, we observe two shifted demand curves, which are reasoned by the typical difference between higher demand at working days and lower demand at weekends or holidays respectively. However, both working day demand and weekend-demand follow the same (weather-dependent) overall annual season.

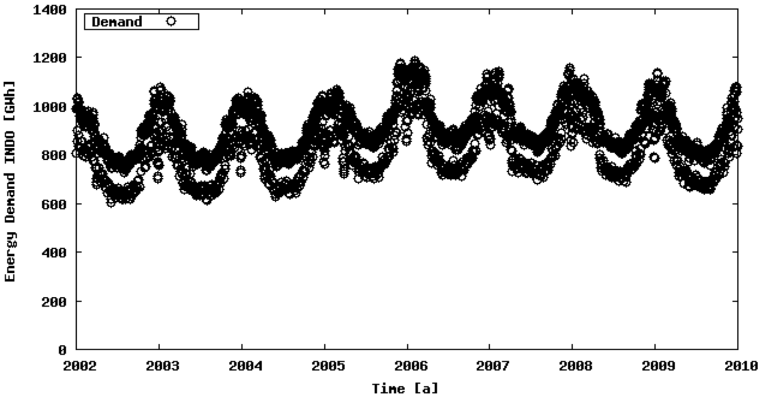
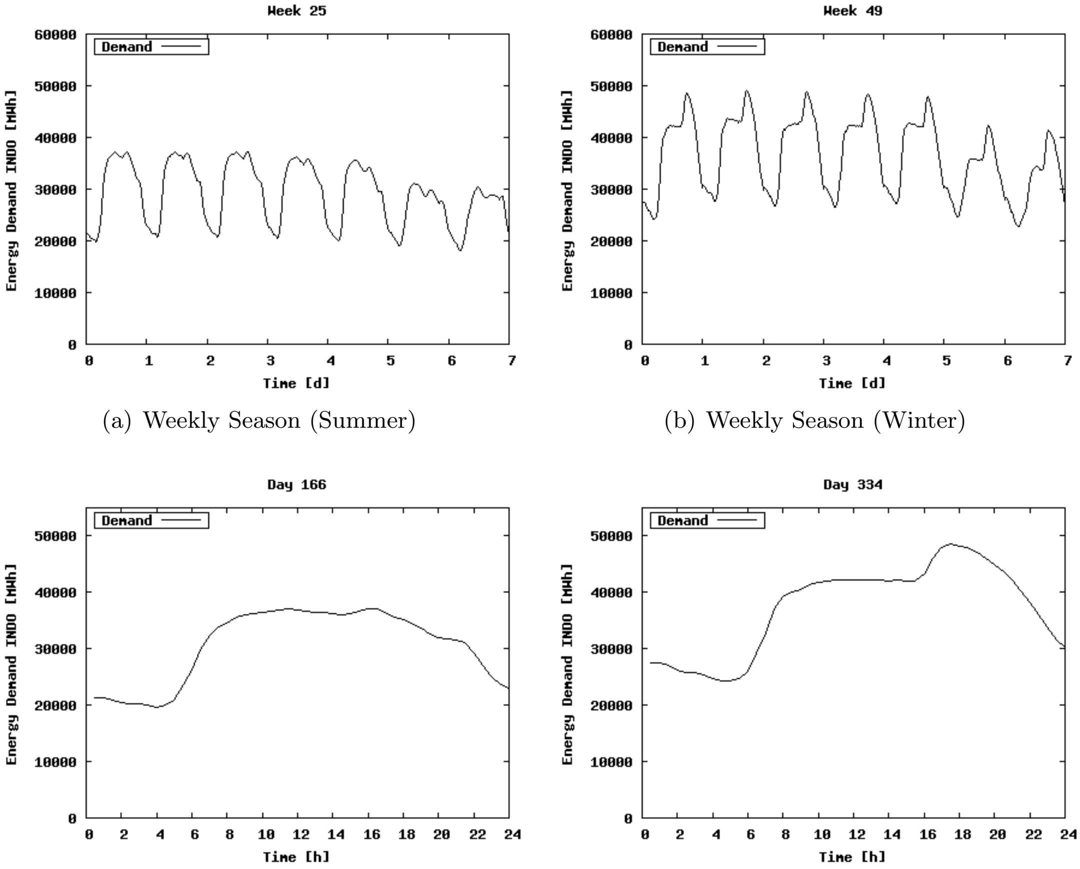


Figure 2.2: Data Set A – Annual Season of Total Energy Demand UK

As already shown, beside the annual season, there are two more seasonal influences. Figure 2.3 illustrates examples for both the weekly season as well as the daily season. First, Figure 2.3(a) and 2.3(b) compare the intra-week season. Essentially, we see that the demand at the weekend is lower than during working days. This difference is independent of annual season. In addition, there is the intra-day season shown in Figure 2.3(c) and 2.3(d), where we observe differences between typical summer and winter days. Both have in common that the demand abruptly increases between 4am and 8am and after that stabilizes at a fairly constant demand. However, in summer days the demand decreases from 4pm to 4am of the next day, while in winter days, there is a peak between 4pm and 10pm and just from 10pm demand decreases until 5am of the next day. In addition, in summer days, the abrupt increase in the morning happens earlier than in winter days.

To summarize, according to the aggregation dimension *time*, there is a weak linear trend over the years, while there are three different seasonal components—namely the



(a) Weekly Season (Summer)

(b) Weekly Season (Winter)

(c) Daily Season (Summer, Monday of Week 25)

(d) Daily Season (Winter, Monday of Week 49)

Figure 2.3: Data Set A – Weakly and Daily Season of Total Energy Demand UK

annual, the weekly and the daily season—that have a strong influence on the total energy demand. In conclusion, especially, the triple seasonality should be taken into account by an appropriate forecast model.

In addition to the trend and seasonality mentioned so far, the predictability of demand and supply also strongly depends on the aggregation level. In general, this is true for all aggregation dimensions such as time, region, customer profile, and product (type of supply or demand). Here, we illustrate this characteristic using the dimensions of region and time.

First, Figure 2.4 illustrates the regional aggregation-dependent predictability. In detail, it shows the energy demand of Germany, Netherlands, Greece, Denmark and Slovenia (the European countries of the MIRACLE partners) as well as the total demand of all countries from 1980 to 2007. Note the log-scaled y-axis. All single demand curves exhibit some fluctuations, while the total demand increases fairly linear. Similar results can be obtained for arbitrary regional subsets and aggregation levels.

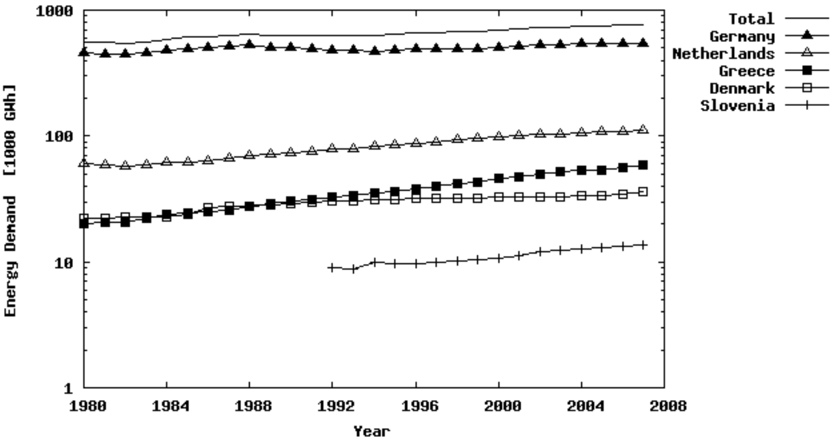
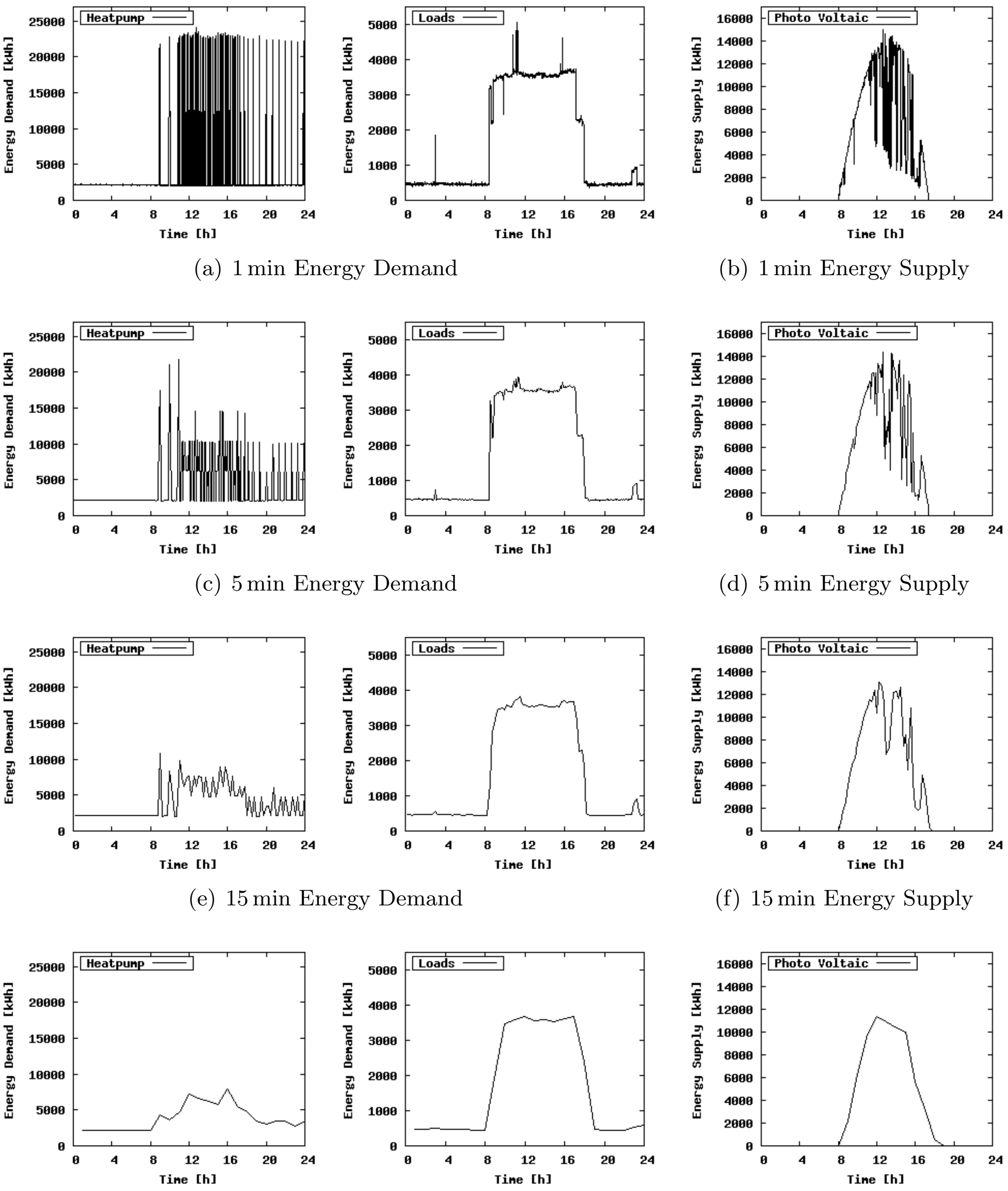


Figure 2.4: Data Set C – Aggregation-Dependent Predictability (Regional)

In conclusion, we can state that the higher the regional aggregation level, the lower the fluctuations. As a result, the higher the aggregation level, the higher the predictability.

As a second example, we use the dimension of time that has typically a huge influence on the predictability. Therefore, we used the CRES data set (household level) of energy demand and supply that exhibits a resolution of 1 min and aggregated the measures over the time. In detail, Figure 2.5 illustrates the measures and aggregates of February 1st 2008 (Friday) of energy demand and supply. Column 1 illustrates the heatpump demand (for cooling and heating), Column 2 shows the additional demand of the building, and Column 3 shows the supply of the photo-voltaic grid. Each column shows, in detail, the influence of increasing the aggregation level. First, the heatpump at 1 min resolution exhibits the characteristics of many short peaks, which is reasoned by fuzzy temperature control. Although, we observe different densities of peaks, only when aggregating to 15 min or hours, we can obtain a predictable time series. Second, also for the other load demand aggregating to 15 min intervals leads to a smoothed time series. Third, the photo-voltaic grid also shows a typical behavior. Clearly, the energy supply is only provided during daytime and the highest supply is achieved at noon. In addition, at this day, there were scattered clouds that led to fine-grained break-downs. When aggregating this series to hours we obtain time series with much better predictability. In addition to the issue of aggregation-dependent predictability, this column shows a further important issue. Namely, the high importance of external input in the form of weather data or even weather forecasts, which are uncertain by itself.

To summarize, it is important to note that there is a high influence of aggregation along the different available dimensions on the predictability of energy demand and supply. Here, especially, the time dimension as well as the integration of external weather data source has huge importance.



(a) 1 min Energy Demand

(b) 1 min Energy Supply

(c) 5 min Energy Demand

(d) 5 min Energy Supply

(e) 15 min Energy Demand

(f) 15 min Energy Supply

(g) 60 min Energy Demand

(h) 60 min Energy Supply

Figure 2.5: Data Set B – Aggregation-Dependent Predictability (Temporal)

3 Time Series Forecasting

There is plenty of existing work about forecasting of time series in general. As a prerequisite for the analysis of related work, in this chapter, we give a broad overview of the mathematical basics of forecasting approaches. The time series forecasting process, in principle, comprises five basic steps as shown in Figure 3.1:

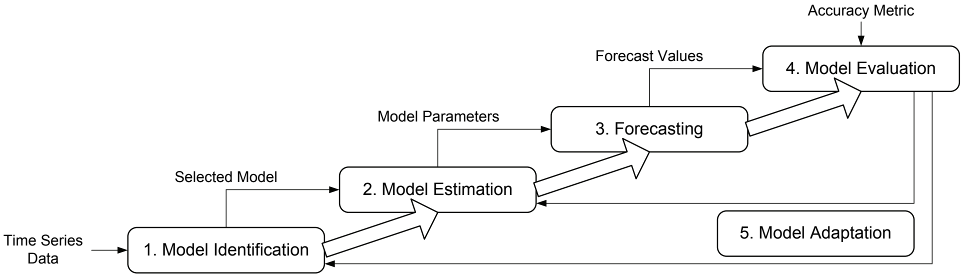


Figure 3.1: Forecasting Process Steps

1. *Model Identification*: It is necessary to choose a forecast model that fits the specific time series best.

2. *Model Estimation*: Each model needs to be configured by estimating its parameters to adapt the model to the specific behavior of a given training time series.

3. *Forecasting*: Forecast future values based on the created forecast model.

4. *Model Evaluation*: Evaluate the forecast model by comparing real observations with forecast values. The accuracy measurement is often accomplished using special accuracy metrics. Besides the possibility to use test data, the model evaluation can also be conducted online in real time as soon as real data becomes available.

5. *Model Adaptation*: Based on the evaluation results, the model has to be adapted by recalculating the parameters or a new model must be chosen better suited for the problem at hand. Similar to the model evaluation, this adaptation can also be conducted online rather than just during the model training.

With regard to the first three steps of this process, we start with the introduction of the two major categories of forecast models. The first category comprises *autoregressive models* introduced by Box et al. [BJR08]. The second category is *exponential smoothing.* In addition, we introduce a third category of approaches that apply machine learning techniques. The resulting overall classification is shown in Figure 3.2. After a detailed introduction of these three basic categories, we describe two common techniques that can be used for parameter estimation, namely Multiple Regression and Maximum Likelihood. In addition, we describe several error metrics that are used to measure the forecast model accuracy, which enables the fourth forecast process step of model evaluation.

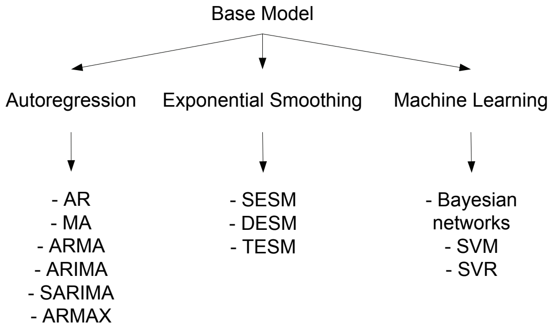


Figure 3.2: Forecast Models Categorized by their Base Approach

The descriptions in this chapter give a short mathematical view on the most common models, techniques and measurements used for time series forecasting. In order to assist the reader in understanding this general introduction to forecasting, we need some basic definitions.

Stochastic Process In probability theory, a stochastic process is the mathematical description of random variables that depend on another parameter (often this parameter is time). In comparison to a deterministic process that does not involve randomness in the development of future states, stochastic processes can evolve in many different directions, even if the initial conditions are known. Stochastic processes can be discrete or continuous, and are described by discrete or continuous probability distributions.

Probability Distribution The probability distribution defines the probability for each possible value of a variable. There are two possible probability distributions. First, the discrete distribution is based on a finite set of values and their dedicated probability (e.g., binomial distribution). Second, the continuous distribution is based on a continuous function describing the probability distribution (e.g., Gaussian distribution).

Probability Density Function The probability density function describes the probability for a continuous variable to have a value within a specific interval.

Probability Mass Function The probability mass function describes the probability for a result to be taken by a discrete variable.

Expected Value The expected value is the mean of all possible values of a variable with respect to their probability distribution. For a discrete variable, it is the sum of all possible values weighted by their probability. For a continuous variable, it is the integral of the possible values weighted using the probability density function of the variable.

Deviation The deviation is the difference between the observed value and the expected value for a variable.

Variance The variance is a measurement of the amount by which the values of a variable may vary. It is calculated as the expectation of the squared deviation of a variable from its mean.

White Noise White noise is a discrete stochastic process of uncorrelated random variables. The expected value is equal to zero and the variance is constant. A special case of white noise is the Gaussian white noise that assumes a normal distribution of the random variables.

Using these general definitions, we now can discuss the three major categories of forecast models in detail.

3.1 Autoregressive Models

Autoregressive models are used to describe the characteristics and the behavior of time series using an auto-regression process. Each model presented below is adapted to interpret specific behavior of time series. Figure 3.3 shows the correlation between the several models and how they are hierarchically connected regarding their containment.

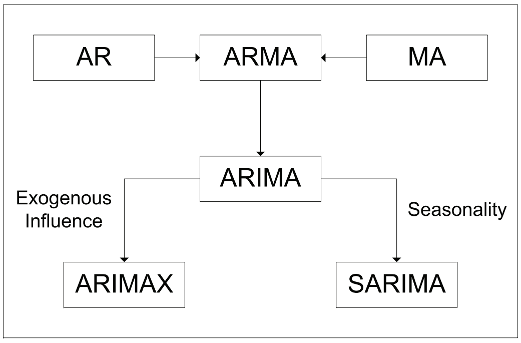


Figure 3.3: Overview of Forecast Models from Box and Jenkins [BJR08]

The meta-models in this section are all models introduced by George Box and Gwilym Jenkins [BJR08]. To ensure the readability of the forecast models, the Backshift Operator has been introduced:

$$BX_t = X_{t-1} \text{ and } B^j X_t = X_{t-j}, \tag{3.1}$$

which maps a value X_t of a time series to its preceding value X_{t-1}. This operator can be used to simplify the notation of a time series model. For example $X_t = a_t - \theta a_{t-1}$, which is a simple Moving Average model and can be written as $X_t = (1 - \theta B)a_t$.

Autoregressive Model (AR)

The autoregressive model [BJR08, BD02] uses a linear combination of previous values of the stochastic process combined with a random shock out of white noise data. The model is described by the following equation:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + ... + \phi_p X_{t-p} + a_t = \sum_{i=1}^{p} \phi_i X_{t-i} + a_t. \tag{3.2}$$

With this equation, the value X_t at time t with the weighted sum of previous values of the AR Process is calculated. Therefore, X_t can be seen as a dependent variable relying on the former output of the AR process. The equation also contains the aforementioned white noise a_t. An AR model is often referred to as AR(p) with p being the order of the Auto Regression process. The parameter describes the number of previous values included in the model. The model parameters ϕ_p can be calculated, for example, using the Least Square Method with the Yule-Walker equations or the Maximum Likelihood Method [Esh09].

Moving Average (MA)

The Moving Average model [BJR08, BD02] describes a time series as a weighted linear combination of several previous values of a white noise process; i.e., a set of uncorrelated, normal-distributed, random variables with an assumed equal variance. In a more straightforward description, the MA model is a regression of the current value of the stochastic process against previous white noise random shocks. The notation of this model is:

$$X_t = a_t + \theta_1 a_{t-1} + \theta_2 a_{t-2} + ... + \theta_q a_{t-q} = a_t + \sum_{i=1}^{q} \theta_i a_{t-i}. \tag{3.3}$$

The values of the white noise process at time t are given as a_t, and $1, -\theta_1, ..., -\theta_q$ are absolutely summable weights, where *absolute summable* refers to the associative relationship of sum and expected value. The MA model is often referred to as MA(q), where q is known as the degree of the MA model. The parameter defines the number of white noise random shocks in the model. The weights of a Moving Average process describing the function cannot be estimated as easily as with the autoregressive model because the random shocks are not directly observable. A nonlinear iterative fitting method is necessary, e.g., Nonlinear Least Square Regression.

Autoregressive Moving Average (ARMA)

The ARMA model [BJR08, BD02] combines the two previously mentioned stochastic models, the Autoregressive model and the Moving Average model. As mentioned by Box and Jenkins [BJR08], there is a strong relationship between both single approaches, because it is possible to transform both models into each other. In addition, the weights used in one of the single models can be determined with known weights of the other one.

In fact, an MA(1) model can be converted to an AR(∞) model and an AR(1) model can be converted to an MA(∞) model.

Most time series can not be described solely by an AR or a MA forecast model, because they show behavior of both models at the same time. For this reason, a combination of both models is useful to describe such time series. Due to the fact that a representation of a finite model of one type can be represented by an infinite representation of its counterpart (huge number of parameters and used lags), AR and MA models can be combined to one single representation called Auto Regressive Moving Average model (ARMA). The ARMA model is described by the following equation:

$$X_t = \sum_{i=1}^{p} \phi_i X_{t-i} + a_t + \sum_{i=1}^{q} \theta_i a_{t-i} \text{ respectively } (1+\sum_{i=1}^{p} \phi_i B^i) X_t = (1+\sum_{i=1}^{q} \theta_i B^q) a_t. \quad (3.4)$$

In many cases, the ARMA model is referred to as ARMA(p, q) where p is the order of the AR part and q is the order of the MA part. The order values p and q can be estimated by analyzing the autocorrelation function (ACF) and the partial autocorrelation function (PACF). In addition, it is possible to create several ARMA models with different values of p and q and to evaluate the accuracy of all model instances. The instance with the lowest average error is chosen to represent the specific time series.

Autoregressive Integrated Moving Average (ARIMA)

With the formerly mentioned models, only stationary time series can be expressed, where *stationarity* refers to a constant joint probability distribution of a stochastic process which implies constant expected value and variance. To express non-stationary time series, the AR part is generalized formulating the ARIMA model [BJR08, BD02]. The model adjusts for the non-stationarity by differencing the respective time series, adding an integrated part. Differencing is a commonly used technique to erase trends and non-stationary behavior. For each value, the differenced value is calculated by subtracting the former value from the actual value:

$$Y_t' = Y_t - Y_{t-1}. \quad (3.5)$$

This method can be used to any order, with the difference of higher order terms involving more former values looks like:

$$Y_t'' = Y_t' - Y_{t-1}' = (Y_t - Y_{t-1}) - (Y_{t-1} - Y_{t-2}). \quad (3.6)$$

The wording *integrated* within ARIMA comes from the fact that the ARIMA model integrates or sums a stationary ARMA process. With the inclusion of the integrated part, the ARIMA model is described by the following equation:

$$(1 + \sum_{i=1}^{p} \phi_i B^i)(1 - B)^d X_t = (1 + \sum_{i=1}^{q} \theta_i B^q) a_t, \quad (3.7)$$

with $(1-B)^d$ being the integrated part of the ARIMA model. The ARIMA model is parametrized by the three parameters p, d, q, and is called ARIMA(p,d,q), with p being the order of the autoregressive part, d being the order of the integrated part and q being the order of the moving average part.

The ARIMA model counts as one of the most general models available for describing time series. All formerly introduced models are in fact special cases of the ARIMA model. Some special cases of the ARIMA model are: ARIMA$(1,0,0)$ = AR(1), ARIMA$(0,1,0)$ = Random Walk, ARIMA$(0,0,1)$ = MA(1), ARIMA$(p,0,q)$ = ARMA(p,q). A common ARIMA model in the economic domain is ARIMA$(1,1,1)$ that can be described as:

$$X_t = X_{t-1} + \phi(X_{t-1} - X_{t-2}) - \theta a_{t-1}. \tag{3.8}$$

Seasonal Autoregressive Integrated Moving Average (SARIMA)

In addition to non-stationary behavior, many time series have a seasonal behavior, which means reoccurring events (peaks, lows) after certain periods in time. This is accounted for by dividing the time series in a seasonal and an unseasonal part. Combining both parts together results in the SARIMA model [BJR08, BD02] or ARIMA$(p,d,q) \times (P,D,Q)_h$. In fact, the SARIMA model is a combination of two ARIMA models, one describing the base time series and the second describing the seasonality. The seasonal part can be explained with an ARIMA model as follows:

$$(1 + \sum_{i=1}^{p} \Phi B^i)^s (1 - B^s)^D X_t = (1 + \sum_{i=1}^{q} \Theta_i B^q)^s a_t, \tag{3.9}$$

with Φ, Θ being the known weights of the components and s being the length of the season (e.g., 12 for an annually reoccurring behavior, when the base unit is a month). This is now combined with the standard ARIMA model, which leads to the following equation:

$$(1+\sum_{i=1}^{p} \phi B^i)(1+\sum_{i=1}^{p} \Phi B^i)^s (1-B)^d (1-B^s)^D X_t = (1+\sum_{i=1}^{q} \theta_i B^q)(1+\sum_{i=1}^{q} \Theta_i B^q)^s a_t. \tag{3.10}$$

Here, all capital letters (Greek and Latin letters) belong to the seasonal part, whereas the lower-case letters represent the non-seasonal part. One model that is often applied in the economic area is the SARIMA$(0,1,1) \times (0,1,1)_{12}$, which describes an annually reoccurring behavior as necessary for many businesses. It can be seen that this model consists of a moving average part and an integrated part but it does not contain an autoregressive part. This model can be represented as

$$(1 - B^{12})(1 - B)X_t = (1 - \theta B)(1 - \theta B^{12}) a_t. \tag{3.11}$$

Autoregressive Moving Average with Exogenous Inputs (ARMAX)

In some cases, a time series is influenced by external variables, which might be described by another time series. The ARMAX model takes this influence into consideration by adding a term for this exogenous data. The ARMAX(p, q, b) then formulates a stationary time series under the influence of another exogenous time series. The model can be described by

$$(1 + \sum_{i=1}^{p} \phi_i B^i) X_t = (1 + \sum_{i=1}^{q} \theta_i B^q) a_t + (1 + \sum_{i=1}^{b} \eta_i B^b) d_t, \tag{3.12}$$

where η_i are the parameters of the exogenous time series d_t and b is the order of this time series. In addition, the ARIMAX(p, d, q, b) model is an extension of the ARMAX model for non-stationary data and can be described as

$$(1 + \sum_{i=1}^{p} \phi_i B^i)(1 - B)^d X_t = (1 + \sum_{i=1}^{q} \theta_i B^q) a_t + (1 + \sum_{i=1}^{b} \eta_i B^b) d_t. \tag{3.13}$$

Additional influencing time series can be added by adding more exogenous terms to the model.

Model Identification for Autoregressive Models

Autocorrelation Function (ACF)

The autocorrelation is the correlation of a time series with itself. In fact, it quantifies the correlation between a set of observations of a time series and a second set of observations of the same time series at a different lag. Each element of the second set of observations has the same lag as the corresponding observation of the first set. In time series analysis, the autocorrelation is used to detect the randomness of data, to find seasonality and trends, and especially, to estimate the right model to be used for a time series. The function describing the autocorrelation between the two sets regarding the lag or the time difference is called autocorrelation function (ACF) [BJR08, NIS10]. The autocorrelation function computes the autocorrelation coefficient that is the measurement of the correlation between two values at a given lag. The autocorrelation coefficient at lag k can be calculated by the following equation:

$$r_k = \frac{\sum_{t=1}^{n-k} (X_t - \bar{X})(X_{t+k} - \bar{X})}{\sum_{t}^{n} (X_t - \bar{X})^2}. \tag{3.14}$$

Here, X_t is the actual value at the index t, X_{t+k} is the value at the index t plus the lag k and $\bar{X}$ is the mean of the values of the time series. Besides this empirical calculation of the autocorrelation coefficient, it can also be determined analytically [BW09].

Partial Autocorrelation Function (PACF)

The partial autocorrelation function (PACF) [BJR08, BD91, NIS10] is also a correlation between different values within the same time series. It measures the correlation between the two variables X_t and X_{t+k} by filtering out linear influences of the variables that lie in between and afterwards calculating the correlation of the transformed random variables.

Thus, it is the autocorrelation between X_t and X_{t+k} that is not accounted for by lags 1 to $k-1$. The PACF identifies the order of an AR model by the fact that the PACF of an AR model is zero at lag $p+1$ and greater. For most time series, the PACF will not be exactly zero but within a certain limit. The PACF can be calculated empirically, for example, using the Yule Walker equations with the Cramer-Rule or using the Durbin-Levinson algorithm [DLL03, Här00, McC98].

Estimating the Model

With the ACF and the PACF, the model to describe a time series correctly can be estimated. In addition, they are used to find the correct order of the model. However, due to the fact that real time series do not behave like perfect autoregressive models, the estimations provided by the ACF and PACF can only be seen as a hint with no guarantee to be correct. Table 3.1 gives an overview over the relation between the time series and the ACFs:

Table 3.1: Overview of the Characteristics of ACF and PACF for Different Models

Models	**Autocorrelation Function**	**Partial Autocorrelation Function**
$AR(p)$	Exponentially or alternating decaying to 0.	Cuts off after some peaks. Gets 0 at order p.
$MA(q)$	Cuts off after some peaks. Gets 0 at order q.	Exponentially or alternating decaying to 0.
$ARMA(p,q)$	Decay starting after a few lags. Starts decaying at order $q-p$.	Decay starting after a few lags. Starts decaying at order $p-q$.
$ARIMA(p,d,q)$	No decay to zero	
Seasonal	High peaks at specific intervals	

3.2 Moving Average and Exponential Smoothing

The moving average technique uses a sliding window of different length over a time series to calculate the average of the data subset within the window. This technique can be used to smoothen erratic time series and to predict future values. There are several options in which way the moving average can be used. In addition to the simple calculation, there are several weighted methods to take the decreased importance of older data points into consideration. Moreover, exponential smoothing, as a moving average technique with exponentially decaying weights, can be used to efficiently describe non-stationary and seasonal behavior.

Simple Moving Average

The simple moving average algorithm takes a certain amount of previous values of a time series and calculates the arithmetic mean. Thus, all values are weighted with constant weights. Further, there is also a centered moving average approach, which takes previous and following values into consideration but this approach is less important in forecasting because no prediction of future values is possible. It is described by

$$\bar{x}_n = \frac{\sum_{i=1}^{j} x_{n-i}}{j}; \; j \in \mathbb{Z}, \tag{3.15}$$

where j describes the length of the sliding window.

Linear Weighted Moving Average

The weighted moving average weights the data points with linear decaying weights. This weighting is done based on the assumption that older data points are less important than more recent ones for the calculation of an actual value. One commonly used instance of this approach is weighting by position. This means that the weight of a value corresponds to its position in the time series. Therefore, the oldest value of a time series is weighted by one, whereas the 15th value is weighted by 15. To compensate for the higher values caused by the weights, the calculation is normalized by the sum of all weights. It is described by

$$\bar{x}_n = \frac{n x_N + (n-1) x_{N-1} + ... + 2x_2 + x_1}{n + (n-1) + ... + 2 + 1}. \tag{3.16}$$

Exponential Moving Average/Exponential Smoothing

Whereas the Linear Weighted Moving Average only uses linear decaying weights, the exponential smoothing uses weights decaying exponentially over time. As with the linear moving average, recent observations are given more weight in relation to older values. The value of the weights can be estimated, yet to get an optimal value it should be calculated. By combining several components the Exponential Smoothing can be enhanced for the description of non-stationary and seasonal time series. This can be achieved by the additive or multiplicative combination of the following components:

- Base Component: Standard component for stationary time series.
- Trend Exponential Smoothing: Enhancement component for time series with a trend.
- Seasonal Exponential Smoothing: Enhancement component for time series with seasonality.

The number of combined components determines the order of the Exponential Smoothing. The most common variants are called *Single*, *Double* and *Triple* Exponential Smoothing.

Single Exponential Smoothing (SESM)

The Single Exponential Smoothing [NIS10] has one weight parameter, also known as the smoothing constant. This method is mainly used for stationary time series fluctuating around a constant mean. The single exponential smoothing is realized with

$$\bar{X}_t = \alpha X_{t-1} + (1-\alpha)\bar{X}_{t-1}. \tag{3.17}$$

The exponential characteristic of this approach can be revealed by a repeated use of the algorithm. The following example equation explains the process:

$$\begin{aligned} \bar{X}_t &= \alpha X_{t-1} + (1-\alpha)[\alpha X_{t-2} + (1-\alpha)\bar{X}_{t-2}] \\ &= \alpha X_{t-1} + \alpha(1-\alpha)X_{t-2} + (1-\alpha)^2\bar{X}_{t-2}. \end{aligned} \tag{3.18}$$

This iterative calculation yields:

$$\bar{X}_t = \alpha \sum_{i=1}^{t-1}(1-\alpha)^{i-1}X_{t-i} + (1-\alpha)^{t-2}\bar{X}_{t-i}. \tag{3.19}$$

It can be seen that the weights decrease exponentially with the time difference $i-1$, which means the contribution of older values decrease in the same fashion. As already mentioned, the value of α can be chosen arbitrarily, but the accuracy of the result highly depends on the chosen α. Therefore, an α should be used that minimizes the Mean Square Error (MSE). One method suggests the calculation of several Exponential Smoothing models with different α and to choose the one with the smallest MSE. Due to the computation overhead caused by this trial-and-error method, other methods are suggested. Examples are non-linear calculation algorithms for this minimization problem namely the Levenberg-Marquardt Algorithm (LMA) or the Gauss-Newton Approach (GMA) [NIS10].

The Forecasting using the Single Exponential Smoothing is done using the Equation 3.17. With this equation, the next single value can be calculated. In most cases more than one value should be predicted. The following equation is used for this purpose:

$$\bar{X}_{t+2} = \bar{X}_{t+1} + \alpha(\epsilon_{t+1}), \tag{3.20}$$

with ϵ being the error of the forecasting calculation. This means that the next forecast value is the old one plus the weighted error of the last one. When no error and no real observations are available, the calculation uses the last known data point plus its predecessor. This is called bootstrapping [NIS10].

Double Exponential Smoothing (DESM)

As mentioned above, the Single Exponential Smoothing does not deliver accurate results for time series containing a trend component. For such time series, the Double Exponential Smoothing technique [NIS10] is used. Double Exponential Smoothing introduces a second smoothing constant β which is chosen in conjunction to α. The technique is described best with two equations that are used in combination:

$$\begin{aligned} \bar{X}_t &= \alpha X_t + (1-\alpha)(\bar{X}_{t-1} + b_{t-1}) \\ b_t &= \beta(\bar{X}_t - \bar{X}_{t-1}) + (1-\beta)b_{t-1}. \end{aligned} \tag{3.21}$$

Forecasting is realized by adding the actual value of both equations:

$$\begin{aligned} \bar{X}_{t+1} &= \bar{X}_t + b_t \\ \bar{X}_{t+m} &= \bar{X}_t + m \cdot b_t, \end{aligned} \tag{3.22}$$

where the m-period ahead forecast is similar to the bootstrapping of the SESM.

Triple Exponential Smoothing (TESM)

Whenever a time series shows a trend and a seasonality, the addition of a third smoothing constant is necessary. This shows that each additional component of a time series results in the addition of a new smoothing constant/equation. This means that for the triple exponential smoothing model [NIS10], one equation describes the smoothing, the second describes the trend and the third describes the seasonal behavior. This leads to the following set of equations:

$$\begin{aligned} \bar{X}_t &= \alpha \frac{X_t}{I_{t-L}} + (1-\alpha)(\bar{X}_{t-1} + b_{t-1}) \\ b_t &= \beta(\bar{X}_t - \bar{X}_{t-1}) + (1-\beta)b_{t-1} \\ I_t &= \gamma \frac{X_t}{\bar{X}_t} + (1-\gamma)I_{t-L}, \end{aligned} \tag{3.23}$$

where b_t is the trend component and I_t is the seasonal component. The index L represents the period of the seasonal component.

While the calculation of the initial values for the smoothing constants of the basic component is rather simple, the calculation for the trend and seasonal component is more complex. The trend needs to be calculated for each period of the season, which means that the smoothing constant needs to be estimated from one period to the next by:

$$b = \frac{1}{L}\left(\frac{X_{L+1} - X_1}{L} + \frac{X_{L+2} - X_2}{L} + ... + \frac{X_{L+L} - X_L}{L}\right). \tag{3.24}$$

The calculation of the smoothing constant for the seasonal component γ is a three-step process:

1. Calculate the average for the available periods:

$$\mathrm{AVG}_s = \sum_{i=1}^{p} X_i, \tag{3.25}$$

with p being the periods of a season and s being the number of considered seasons.

2. Divide the observations through the respective mean and create a matrix with the values:
$$\begin{pmatrix} X_1/A_1 & X_{p+1}/A_2 & ... & X_{sp-p+1}/A_s \\ X_2/A_1 & X_{p+2}/A_2 & & X_{sp-p+2}/A_s \\ \vdots & \vdots & ... & \\ X_p/A_1 & X_{2p}/A_2 & & X_{sp}/A_s \end{pmatrix}. \tag{3.26}$$

3. Read the rows of the matrix and calculate the average of each row. The result of this calculation provides the value of the different instances of the seasonal component smoothing constant:
$$I_p = (X_p/A_1 + X_{2p}/A_2 + ... + X_{sp}/A_s)/s. \tag{3.27}$$

The forecasting for the TESM can be calculated by:

$$\bar{X}_{t+m} = (\bar{X}_t + m \cdot b_t) I_{t-L+m}. \tag{3.28}$$

The calculation can be done in additive and multiplicative fashion, which means the exchange of the arithmetic operators + and ·.

The Moving Average and Exponential Smoothing techniques are commonly used approaches for time series forecasting. Even non-stationary and seasonal time series can be described with special versions of the Exponential Smoothing technique.

3.3 Machine Learning Techniques

In addition to autoregressive models or exponential smoothing, approaches from the research area of machine learning are used to solve the forecast problem as well. In this section, we describe the mathematical basics of these machine learning techniques.

Bayesian Networks (BN)

Bayesian Networks are decision networks used to represent knowledge about an uncertain domain. The domains are typically represented as an n-dimensional set of stochastic observations, which can be correlated among each other; much like time series. Bayesian Networks create a high-dimensional joint probability distribution based on local probabilistic rules. Therefore, knowledge about the probabilities is necessary to create a BN. A common example for BNs is the relationship between symptoms and diseases. The symptoms are the set of variables and a BN describes the relationship between the occurrence of certain symptoms and the probability of the presence of certain diseases.

A BN is described by the following characteristics:

- Vertices V: represent a set of stochastic variables,
- Directed edges E: describe the direct influence of one variable to another,
- No cycles: a BN is a directed acyclic graph (DAG),

- A complete conditional probability table for each vertex describing the dependency of the predecessor to its successor, and
- Specifications of the correlations: vertices without predecessor need an initial probability $P(A = i)$ and vertices with predecessor need conditional probabilities $P(A = i \,|\, B = j, C = k)$.

A BN with these properties can formally be described as $BN = (G, \Theta)$, where $G = (V, E)$ stands for the directed acyclic graph with the set of nodes that represent the variables and the edges that represent the dependencies between the variables. Each variable of G is independent of its non-descendants. The Θ is a set of parameters of the graph. In this set the parameter θ is contained for each observation of the variable of G. The probability of an observation is given by θ and is conditioned on the set of parents for a variable. Variables without a parent are said to be unconditional. Formally, this is written as $\theta_{x_i \,|\, \pi_i} = P_B(x_i \,|\, \pi_i)$ with π_i being the set of parents of a specific variable. Therefore, the joint probability distribution of G is defined as:

$$P_B(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} P_B(X_i \,|\, \pi_i) = \prod_{i=1}^{n} \theta_{x_i \,|\, \pi_i} \tag{3.29}$$

In addition to this modeling of the BN, which assumes that the variables are discrete, a Gaussian modeling of the Bayesian network is also possible. The Gaussian modeling assumes that the variables have a multivariate normal distribution [CSG04].

The conditional dependencies between variables are often estimated using known statistical and computational methods; e.g., the recovery algorithm developed by Rebane and Pearl [RP87]. The reasoning with the help of a Bayesian network can now be done by combining the separate local probabilities and following the path from the top to the bottom. The way the joint distribution is calculated is different from just following the path and factoring up the probabilities. Due to the statement of independences, which says that each variable is independent of its non-descendants, the joint distribution of the nodes is not calculated by the chain rule. Instead the BN defines a unique joint distribution in a form that just factors the direct dependencies of the variables. The following example explains the joint distribution using direct dependencies.

Example 1. *Figure 3.4 illustrates a simple representation of a Bayesian network without the conditional probability tables.*

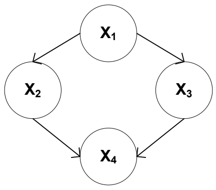


Figure 3.4: Example Bayesian Net without Conditional Probability Tables

Using the chain rule, the joint distribution is:

$$P_B(X_1, X_2, X_3, X_4) = P(X_1) \cdot P(X_2 \,|\, X_1) \cdot P(X_3 \,|\, X_1) \cdot P(X_4 \,|\, X_1, X_2, X_3)$$
$$P_B(X_1, X_2, X_3, X_4) = P(X_1) \cdot P(X_2 \,|\, X_1) \cdot P(X_3 \,|\, X_1) \cdot P(X_4 \,|\, X_2, X_3),$$

where the first equation was reduced by using the statement of independent relationships. This simplifies the BN probability calculation, especially for large BNs.

The creation of a Bayesian network comprises the following steps:

1. Determine a set of relevant variables describing the process and/or the domain.
2. Create an order of the variables observing the causal influence (causes are always predecessors of consequences).
3. Take a variable and create a node.
4. Determine a minimal set of dependencies to all nodes already in the BN watching the conditional independence.
5. Determine the conditional probability table for the variable.
6. Repeat from step 3 as long as there are variables.

The computational creation of a BN is a non-trivial task. However, the created BN enables reasoning.

There are several possibilities for inference within BNs such as *Diagnostic Inference* (meaning to reason from consequences to the causes), *Causal Inference* (meaning to reason from causes to the consequences), *Intercausal Inference* (meaning to reason between several causes having a common consequence), and *Combined Inference* (meaning a combination of the above mentioned inferences). The main goal of inference in Bayesian networks is to estimate the values of hidden nodes, given the values of observed nodes. This ability of Bayesian networks is used in the forecasting domain to predict future values. The BN models the dependencies between different variables of a time series. The forecasting can then be seen as an inference problem of the BN. It partitions the corresponding variables and estimates the optimal value of a variable using the minimum of the mean squared error [ZSY04]. For example, considering two variables x_E and x_F, the expected value (E) for x_F given x_E using the minimum mean square error is:

$$\hat{x}_F = E(x_F \,|\, x_E).$$

Bayesian networks are applied to weather forecasts [CSG04, CCSG02] and also used in other domains, such as traffic flow forecasting [ZSY04] and energy load forecasting [CS03].

Support Vector Machines (SVM)

A Support Vector Machine is an approach in the field of machine learning for classification, pattern recognition and regression. In the simplest case, an SVM is used to divide a set of data points into two classes. The goal thereby is to find a dividing classifier that maximizes the distance between the classes. This classifier is called hyperplane. The data of a data set is divided by the hyperplane into two categories. There might be several classifiers that are able to divide the given data but only one produces a maximum margin between the hyperplane and the data points. This particular classifier is called the optimal separating hyperplane that is the basis of the support vector machine. The simple binary case can be described by:

$$X = \{(x_i, y_i) | x_i \in \mathbb{R}^p, y_i \in \{-1, 1\}\}_{i=1}^{n}, \tag{3.30}$$

where each x_i is a p-dimensional vector containing the values that is called input and y_i is called the output and describes the resulting categories and assignment of data points.

The hyperplane that divides those variables x_i into several categories is described by:

- A normal vector w.
- A parameter $\frac{b}{||w||}$ that describes the perpendicular distance from the hyperplane to the origin.

Given those two definitions the hyperplane is defined as:

$$w \cdot x - b = 0 \tag{3.31}$$

where $\cdot$ stands for the scalar product. The optimization problem is now to select the parameters w and b in a way that all data points can be described by:

$$\begin{aligned} w \cdot x_i - b &\geq +1 \text{ for } y_i = +1 \\ w \cdot x_i - b &\leq -1 \text{ for } y_i = -1 \end{aligned} \tag{3.32}$$

The creation of the hyperplane is based on just a subset of the data set instead of the full set. This subset is called the *support vectors*. Support vectors are the points nearest to the hyperplane and they determine its behavior. Points that are further away do not have any influence. Using the condition given by Equation 3.32, we find two additional hyperplanes the support vectors lie on. They are defined as:

$$\begin{aligned} w \cdot x - b &= +1 \\ w \cdot x - b &= -1 \end{aligned} \tag{3.33}$$

The distance between the classifying hyperplane and the support vector hyperplanes is equidistant and called the margin of the SVM. In a linear data environment, the data can be separated by a linear hyperplane for which the margin between both support vector hyperplanes can be calculated by $\frac{2}{||w||}$. The maximization problem for the margin

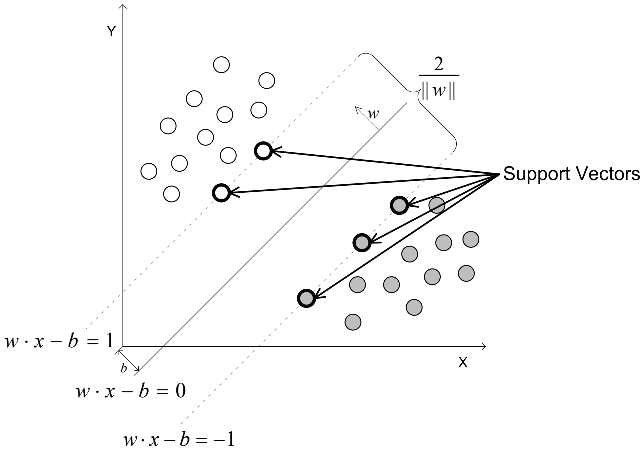


Figure 3.5: Classification with an SVM

therefore comprises the minimization of $||w||$. The SVM classification using a hyperplane is illustrated in Figure 3.5.

The maximization problem is subject to the condition that no data points fall in the margin between the two support vector hyperplanes described by Equation 3.33. This constraint can be formulated as

$$y_i(w \cdot x_i - b) \geq 1, \tag{3.34}$$

by combining the conditions from Equation 3.32. As this constraint is accomplished by the support vectors we can calculate b from:

$$b = \frac{1}{n_{sv}} \sum_{i=1}^{n_{sv}} (w \cdot x_i - y_i), \tag{3.35}$$

where n_{sv} is the number of Support Vectors available.

Since the norm of w contains a square root, $||w||$ is often substituted by $\frac{1}{2}||w||^2$, yet this does not change the solution. Due to the optimization problem with additional constraints, Lagrange Multipliers have to be allocated. Those multipliers are named α and are given by:

$$L(\alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{L} \alpha_i y_i (w \cdot x_i - b) + \sum_{i=1}^{L} \alpha_i. \tag{3.36}$$

To solve this equation, it is necessary to find the values of w and b that minimize this equation and α that maximizes it. It is necessary to maximize α to ensure that the best hyperplane is selected and not a set of hyperplanes fulfilling the constraints. This is a classical maximization/minimization that can be solved by differentiating and setting the derivatives to zero:

$$\frac{d\,L(\alpha)}{d\,w} = 0 \rightarrow w = \sum_{i=1}^{L} \alpha_i y_i x_i \text{ and } \frac{d\,L(\alpha)}{d\,b} = 0 \rightarrow \sum_{i=1}^{L} \alpha_i y_i = 0. \tag{3.37}$$

This already constitutes a solution for w, which can be substituted into the equation of $L(\alpha)$. This leads to:

$$L_D(\alpha) = \sum_{i=1}^{L} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i x_j \quad | \quad \alpha_i \geq 0, \sum_{i=1}^{L} \alpha_i y_i = 0. \tag{3.38}$$

This can be simplified by introducing the so called Hessian $H_{ij} = y_i y_j x_i x_j$. In addition, a second Lagrange Multiplier λ is introduced to meet the constraint given by the second derivative. This leads to

$$L_D(\alpha) = \sum_{i=1}^{L} \alpha_i - \frac{1}{2} \alpha^T \mathbf{H} \alpha \quad | \quad \alpha_i \geq 0, \sum_{i=1}^{L} \alpha_i y_i = 0. \tag{3.39}$$

The D within L_D stands for the optimization problem in its dual form. This problem and especially the value of α can now be determined using standard quadratic programming techniques and programs. Detailed information can be found in [Fle08, Bur98, Gun98]. With determined α, w and b can be calculated and the optimization problem can be solved.

In the original form of an SVM, only binary classification is possible. However, in general, an SVM is not limited to a binary deviation of the variables, since many hyperplanes can be placed to divide the data. In addition, it is not always possible to divide data in a linear way. For this reason a non-linear classifying SVM exist, which comprises the kernel trick substituting scalar products by kernel functions. Details for non-linear SVMs and examples for kernel functions can be found in [Fle08, Bur98, Gun98, SS98].

Support Vector Regression

The previous subsection describes the SVM in a common, mathematical way. The outcome of a standard SVM is a division of data into several classes. Beside the classification possibilities of the SVM, it can be also used for regression. The basic idea of support vector regression is to map the data from a time series to a high-dimensional feature space and to perform linear regression in this space. Thus, linear regression in a high-dimensional space corresponds to nonlinear regression in the low-dimensional space [MSR+97]. To do this, an adaptation of the SVM technique is necessary. Instead of just sorting input values into two categories, a real value prediction for y_i is necessary. The training data is therefore now formulated as

$$\{x_i, y_i\} \text{ where } i = 1...L,\ y_i \in \mathbb{R},\ x \in \mathbb{R}^D,\ y_i = w \cdot x_i + b. \tag{3.40}$$

The SVR minimizes a risk function which describes the penalization of errors in the time series. This regulated risk function is described as:

$$R_{reg}[f] := R_{emp}[f] + \lambda ||w||^2 = \sum_{i=1}^{l} C(f(x_i) - y_i) + \lambda ||w||^2 \tag{3.41}$$

with R_{emp} being the empirical risk function that needs to be determined. The λ in the equation is the regularization component, which can be determined using several regularization networks, where the approximation function has to be constrained to an appropriately small hypothesis space (a space of machines or networks) [Vap98]. In addition, a cost function C is introduced in the empirical risk function that describes how errors are penalized when being above a specific threshold. The standard choice for this cost function is:

$$C(y, x, f(x)) = |y - f(x)|_\epsilon. \tag{3.42}$$

The cost functions are also called loss function. The standard loss function is not applicable in all cases as the linear increase may cause a loss in robustness. Therefore, more sophisticated loss functions have to be applied [Fle08, Gun98, MSR$^+$97]. Examples for such loss functions are: quadratic (which corresponds to the least square method), Laplace (which is more tolerant against outliers), Huber (which is a robust function if the underlying data is unknown) and ϵ- insensitive (which creates a sparse set of vectors). Once a loss function has been defined, the computation is done as a Dual Lagrange optimization problem and is, therefore, solved similarly with as with the standard SVM. The outcome is a function that can be used to predict future values. Figure 3.6 illustrates the concept of forecasting using support vector regression.

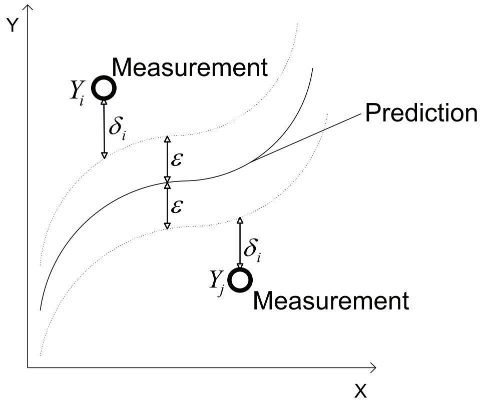


Figure 3.6: Prediction with Support Vector Regression [Kec01]

The whole SVR algorithm can be described as follows:

1. Manually determine a loss function describing the punishment for errors being out of the loss region.

2. Choose suitable parameter values for the loss function (C) and the size of the insensitive loss region (ϵ).

3. Find the Lagrange Coefficients to find a maximum solution for the optimization problem. This can be achieved using a solver for a quadratic problem.

4. Calculate w by means of the equation:

$$w = \sum_{i=1}^{L} (\alpha_i^+ - \alpha_i^-) x_i. \tag{3.43}$$

5. Determine the support vectors by evaluating the constraint $0 < \alpha < C$.

6. Calculate b as follows:

$$b = \frac{1}{n_s} \sum_{s \in S} \left[t_i - \epsilon - \sum_{m=1}^{L} (a_i^+ - \alpha_i^-) x_i \cdot c_m \right]. \tag{3.44}$$

7. A new point $\hat{x}$ can be calculated by evaluating:

$$\hat{y} = \sum_{i=1}^{L} (\alpha_i^+ - \alpha_i^-) x_i \cdot \hat{x} + b. \tag{3.45}$$

The whole approach of Support Vector Regression is described in detail by Mueller et al. [MSR+97], where they successfully applied Support Vector Regression to time series. Further descriptions and research in this field are available [Fle08, Gun98, SS98]. In addition, Pai et al. [PH05] applied Support Vector Machines to forecasting in the energy domain.

Finally, it is important to note that there exist several other machine learning techniques that are, similarly as the described techniques, applied for time series forecasting as well. Examples for these techniques are Artificial Neuronal Networks (ANN), Case-Based Reasoning (CBR), and Symbolic Machine Learning (e.g., Decision Trees and Regression Trees) [Nak94].

3.4 Methods for Parameter Estimation

Time series forecasting can be realized using the aforementioned forecast models. However, additional techniques are necessary in order to create these models and estimate the necessary parameters. Typically, least squares and maximum likelihood estimation are used for this purpose [MEK06]. Therefore, in this section, we describe the mathematical basics of linear regression (as an example for the application of the method of least squares) and the maximum likelihood method.

Multivariate Linear Regression (MLR)

The linear regression algorithm is used to calculate a function by minimizing an error metric. The function is calculated for a variable Y depending on a set of independent variables X.

Simple Linear Regression

Simple Linear Regression uses a function of order one, which means that the calculation is done for one variable x and one variable y. This function is described by:

$$\begin{aligned} y_i &= \alpha + \beta x_i + e_i \\ \tilde{y}_i &= \alpha + \beta x_i \text{ with } e_i = y_i - \tilde{y}_i, \end{aligned} \tag{3.46}$$

where e_i describes the difference between the regression line $\alpha + \beta x_i$ and the measured values y_i. In addition $\tilde{y}_i$ is the estimated value for y_i. The goal of the Least Square Method is to minimize the quadratic error of the function residuals:

$$\min(\alpha, \beta) \left(\sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - (\alpha + \beta x_i))^2 \right). \tag{3.47}$$

Through partial differentiation with respect to α and β, one finds:

$$\beta = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \text{ and } \alpha = \bar{y} - \beta \bar{x}. \tag{3.48}$$

where $\bar{x} = \frac{1}{n} \sum x_i$ and $\bar{y} = \frac{1}{n} \sum y_i$ are the the mean values of x and y, respectively. Thus, with α and β the regression function 3.46 can be determined.

Polynomial Linear Regression

The extension of the simple Linear Regression is the Polynomial Linear Regression described by

$$y = \alpha + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + ... + \beta_n x^n + \epsilon. \tag{3.49}$$

To some extent, the Polynomial Linear Regression can be seen as a special case of the Multiple Linear Regression described below. Therefore, the calculation of the Regression Coefficients α and β is performed in the same way.

Multiple Linear Regression

The simple Linear Regression can be extended to the Multiple Linear Regression which enables (1) polynomials of higher degree, and (2) the consideration of more independent variables. The calculation is conveniently done using a matrix notation. The multiple Linear Regression exploits the direct, linear correlation of y to several variables x. Therefore, to calculate the function, all of these variables are taken into account by

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p + \epsilon. \tag{3.50}$$

Observing that each variable x and Y can have multiple values:

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip} + \epsilon_i, \tag{3.51}$$

one arrives at the linear set of equations:

$$Y = X\beta + \epsilon, \tag{3.52}$$

where we introduced the n-dimensional vectors

$$\begin{aligned} Y &= (Y_1, Y_2, ..., Y_i, ..., Y_n)^T \\ \beta &= (\beta_0, \beta_1, \beta_2, ..., \beta_j, ..., \beta_p)^T \\ \epsilon &= (\epsilon_1, \epsilon_2, ..., \epsilon_i, ..., \epsilon_n)^T \end{aligned} \tag{3.53}$$

and the $n \cdot (p+1)$ dimensional matrix.

$$X = \begin{pmatrix} 1 & x_{11} & \dots & x_{1j} & \dots & x_{1p} \\ 1 & x_{21} & \dots & x_{2j} & \dots & x_{2p} \\ \vdots & & \ddots & & \ddots & \vdots \\ 1 & x_{i1} & \dots & x_{ij} & \dots & x_{ip} \\ \vdots & & \ddots & & \ddots & \vdots \\ 1 & x_{n2} & \dots & x_{nj} & \dots & x_{np} \end{pmatrix}. \tag{3.54}$$

The vector of dependent variables is denoted by Y, β denotes a vector of regression coefficients, ϵ denotes a vector of error coefficients, and X is the matrix of independent variables. Using the method of least squares the Equation 3.52 can be solved by estimating β. The equation for solving β simplified by matrix multiplication is:

$$\beta = (\beta_0, \beta_1, \beta_2, ..., \beta_j, ..., \beta_p)^T = (X^T X)^{-1} X^T Y. \tag{3.55}$$

With determined β, the Equation 3.52 can be solved.

Maximum Likelihood (ML) Function

The Maximum Likelihood Function [BJR08, Pru04, Sto07a, Sto07b] is another method that can be used for the estimation of forecast model parameters. The basic functionality of the ML is an estimation under which circumstances a certain event is most likely to occur. For a given sample of a data set, the probability of the occurrence of the event $Y = y_i$,$i = 1, ..., n$ can be calculated by

$$f(Y;\theta) = \prod_{i=1}^{n} f(Y = y_i;\theta), \tag{3.56}$$

where f is the probability density function. The probability depends on the parameter vector θ, which describes the basic population; e.g., the mean. The ML does not have a given θ but determines the θ were the likelihood L is at the maximum. This means that θ is chosen according to the probability distribution of observed data. The ML function is described as:

$$L(\theta;Y) = \prod_{i=1}^{n} f(\theta;Y). \tag{3.57}$$

We assume that the probability mass function and the specific sample are known. The calculation of the ML is a simple maximum calculation using the derivative of first and second order:

$$\frac{dL}{d\theta} = 0 \text{ and } \frac{d^2L}{d\theta^2} < 0. \tag{3.58}$$

Based on the fact that most probability density functions comprise an exponential component, often the Log-Likelihood-Function is used:

$$\ln L(\theta;\ Y) = \ln[f(y_1, \theta)] + \ln[f(y_2, \theta] + ... + \ln[f(y_n, \theta)] = \sum_{i=1}^{n} \ln[f(y_i, \theta)]. \tag{3.59}$$

To solve the maximization problem, the first and the second order derivations of the Log-Likelihood-Function need to be created and the parameters can be calculated accordingly. The following example explains the use of the Maximum Likelihood Method.

Example 2. *A normal distribution is assumed and the mean and the variance of the data set, using some sample data, are calculated. The probability density function for a normal distributed variable Y is:*

$$f(y_i, \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-((y_i-\mu)^2/(2\sigma^2))}. \tag{3.60}$$

Therefore, the ML is:

$$L(\mu, \sigma^2|y) = \prod_{i=1}^{n} \left\{ \frac{1}{\sigma\sqrt{2\pi}} e^{-((y_i-\mu)^2/(2\sigma^2))} \right\} = \left[\frac{1}{\sigma\sqrt{2\pi}} \right]^n \cdot e^{-(\sum(y_i-\mu)^2)/(2\sigma^2)}. \tag{3.61}$$

Applying the logarithm to simplify this equation, one arrives at:

$$\ln L = -n \ln \sigma - \ln \sqrt{2\pi} - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \mu)^2. \tag{3.62}$$

The mean μ and the variance σ can be calculated by partial differentiation, which leads to the result:

$$\hat{\mu} = \bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i \text{ and } \hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{\mu})^2. \tag{3.63}$$

Thus, the ML can be used to calculate parameters of sample data sets.

ML for the Multivariate Regression

In this subsection, we describe how to use the Maximum Likelihood Method in conjunction with the multivariate regression. The probability density function for a multivariate regression over normally distributed variables is obtained as the generalization of Equation 3.60:

$$f(\epsilon; \beta; \sigma_u^2) = \frac{1}{(2\pi\sigma_u^2)^{n/2}} \cdot e^{-\epsilon^T \epsilon / 2\sigma_u^2}. \tag{3.64}$$

Here, ϵ is the vector of error coefficients. Using Equation 3.52 $Y = X\beta + \epsilon$, ϵ can be calculated by $\epsilon = Y - X\beta$ and thus, Equation 3.64 can be reformulated to:

$$f(y \mid X) = \frac{1}{(2\pi\sigma_u^2)^{n/2}} \cdot e^{-[(y-X\beta)^T(y-X\beta)/2\sigma_u^2]}, \tag{3.65}$$

which describes the conditional probability of y given a value X. The ML regression determines the maximum likelihood L with respect to the regression coefficient β and the variance σ^2. It is again convenient to work with the log-likelihood function:

$$\ln(L) = -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\sigma_u^2) - \frac{1}{2\sigma_u^2} - \frac{1}{2\sigma_u^2}[(y - X\beta)^T(y - X\beta)]. \tag{3.66}$$

The calculation of the regression coefficient and the variance is again a standard maximization problem, leading to:

$$\begin{aligned} \frac{d\ \ln L}{d\ \beta} &= -\frac{1}{2\sigma_u^2}\frac{d\ [(y - X\beta)^T(y - X\beta)]}{d\ \beta} = 0, \\ \frac{d\ \ln L}{d\ \sigma_u^2} &= -\frac{n}{2\sigma_u^2} + \frac{1}{2\sigma_u^4}[(y - X\beta)^T(y - X\beta)] = 0. \end{aligned} \tag{3.67}$$

The first of these equations yields the solution

$$\beta = (X^T X)^{-1} X^T y \tag{3.68}$$

found in Equation 3.55 with the least square method. In this case, the outcome of the ML calculation of the regression coefficient leads to the same equation as the least square method. This only happens when the probability distribution is a normal distribution with expected value of zero and constant variance.

3.5 Measuring the Accuracy

As already mentioned, the accuracy of estimated forecast models is evaluated using certain error metrics. For this purpose, different metrics can be applied according to a given optimization objective. In this section, we describe the most commonly used error metrics.

Mean Square Error (MSE)

The Mean Square Error (MSE) [BJR08, Hyn06] is one of the most commonly used techniques to measure the accuracy of predictions. It is also the basis for the sum of least square approaches used in the linear regression technique. It is the square of the difference of the actual value and the predicted value:

$$\text{MSE} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{3.69}$$

with y_i being the actual value and $\hat{y}_i$ being the predicted value. The smaller the MSE value, the better is the predictive model. A disadvantage of the MSE is that it heavily weights extreme distortions, which means that larger errors have a much larger input than small ones due to the squared error term. To solve the heavy weight of large errors the Normalized MSE (NMSE) has been introduced. It normalizes the above presented equation with the square of the actual value and is described by

$$\text{NMSE} = \sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{y_i^2}. \tag{3.70}$$

Another disadvantage is that the MSE value does not have the unit of the measured time series but the square of it. This is resolved by the Root Means Square Error (RMSE) that is the square root of the MSE and thus, the original unit is used. The RMSE is described by

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}. \tag{3.71}$$

Furthermore, there is a normalized RMSE (NRMSE) that normalizes the RMSE, dividing it by the range of values:

$$\text{NRMSE} = \frac{\text{RMSE}}{x_{max} - x_{min}}. \tag{3.72}$$

Finally, it is important to note that all error calculations that rely on the MSE are not suitable for the comparison of several time series because a larger variance and the increase of the bias leads to an increase of the MSE. In addition, the MSE is symmetric, which means that the direction of the error is not taken into account.

Mean Absolute Error (MAE)

The Mean Absolute Error (MAE) [BJR08, Hyn06] represents a measurement for absolute errors between real values and forecasts. This is done by computing the sum of all differences between real values and predicted values. In detail, the MAE is described by:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} \left| \hat{y}_i - y_i \right|, \tag{3.73}$$

with $\hat{y}_i$ being the forecast value and y_i being the actual value. The MAE is always on the same scale as the data it is calculated from. The values are therefore scale-dependent. This measurement is therefore inappropriate for comparing several different time series.

Mean Absolute Percentage Error (MAPE)

The Mean Absolute Percentage Error (MAPE) [BJR08, Hyn06] estimates the fit of a model by expressing its accuracy as a percentage. The advantage of this accuracy measure is that it is not fixed to a specific unit. Therefore, arbitrary models can be compared regardless of the unit of their values or their level. The MAPE is calculated as the sum of the absolute errors, normalized by the actual value:

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|, \tag{3.74}$$

with $y_i$being the actual value and $\hat{y}_i$ being the predicted value.

The major disadvantage of this error metric is that the MAPE has no upper bound. There is only a lower bound, which is zero. Due to this missing upper bound, extremely high values for certain time series distort the comparability of the MAPE. To adjust for this disadvantage, the adjusted MAPE or symmetric MAPE (SMAPE) is used.

Symmetric Mean Absolute Percentage Error (SMAPE)

The Symmetric MAPE (SMAPE) [BJR08, Hyn06, GL99] enhances the classical MAPE with an upper bound and therefore, it ensures the comparability of the time series. The SMAPE also uses the sum of the absolute errors but normalizes them with the half of the sum of the actual values plus the forecast values. In detail, the SMAPE is described by

$$\text{SMAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{(y_i - \hat{y}_i)/2}. \tag{3.75}$$

The only problem with the SMAPE is that it is not really symmetric because positive and negative errors of the same magnitude resolve in unequal SMAPE values. In extreme cases, even the monotonic relationship between the SMAPE and the absolute error is not given.

Mean Absolute Scaled Error (MASE)

The mean absolute scaled error (MASE) [BJR08, Hyn06] is a generally applicable and comparable measurement for forecast accuracy. It is based on the MAE error metric and is normalized based on the mean MAE of the sample forecast. This means that the MAE is scaled by a one-step-ahead forecast from each data point in the sample. The MASE is defined with:

$$MASE = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\hat{y}_i - y_i}{\frac{1}{n-1}\sum_{i=2}^{n}|y_i - y_{i-1}|}\right). \tag{3.76}$$

The result of the calculation does not depend on the scale of the time series anymore. It is less than one whenever it is conducted from a forecast that is better than the average, naïve one-step forecast. It therefore can be used for the comparison of different time series.

Additional Quality Metric

In addition to the mentioned accuracy measurements that describe the error between real and forecast values, there are additional quality metrics that allow for a more fine-grained analysis forecasting accuracy. As examples for this category, we discuss the Coefficient of Determination as well as the Prediction on Chance in Direction (POCID).

Coefficient of Determination

The Coefficient of Determination describes the variability/variance of the data set that the statistical model takes into account. It is also possible to determine the correlation between the dependent variable (Y_i) and the independent variables (X_i). The Coefficient of Determination can be seen as a measure of how well future values can be predicted by the statistical model. The value of the coefficient lies between 0 and 1, whereas 0 means no linear correlation and 1 means absolute linear correlation. It is named as R^2 accounting for the quadratic calculation and it is described as

$$R^2 = 1 - \frac{\sum_i^n (y_i - \hat{y}_i)^2}{\sum_i^n (y_i - \bar{y})^2}. \tag{3.77}$$

There are advanced definitions of the Coefficient of Determination called Adjusted R^2 and Generalized R^2. The Adjusted R^2 tries to solve the problem that the value of R^2 increases as the number of variables increases, no matter whether or not they increase the quality of the model. The Adjusted R^2 is calculated by:

$$\bar{R}^2 = 1 - (1 - R^2)\frac{n-1}{n-p-1} \tag{3.78}$$

with n being the number of variables in the sample and p being the number of total available variables.

The generalized R^2 has been proposed by Cox and Snell. It is not based on a linear regression but on the maximum likelihood function. It is described by the following equation:

$$R^2 = 1 - \left\{ \frac{L(0)}{\hat{L}(\beta)} \right\}^{\frac{2}{n}}, \tag{3.79}$$

where $L(0)$ is the likelihood of the intersection and $\hat{L}(\beta)$ is the likelihood of the specified model. The number of elements within the sample is depicted by N. All these coefficients describe the quality of a model but do not allow for comparing different forecast models.

Prediction on Chance in Direction (POCID)

The POCID defines a measurement of the number of correct decision whether a value increases or decreases in the next time interval. The POCID can therefore be used to measure the predictive quality of a forecasting (whether the forecasting correctly predicted the increase of a value). The POCID is defined as:

$$\text{POCID} = 100 \cdot \frac{\sum_{i=1}^{n} D_i}{n} \text{ where } D_i \text{ is: } D_i = \begin{cases} 1, & if(\hat{y}_i - \hat{y}_{i-1})(y_i - y_{i-1}) > 0 \\ 0, & \text{otherwise,} \end{cases} \tag{3.80}$$

and thus, lies between 0 and 100. The closer the value is to 100, the better is the prediction of the model [NCR09].

3.6 Summary and Discussion

Essentially, in Chapter 3, we gave a general overview of time series forecasting. In detail, we classified forecast models into the three categories of (1) autoregressive models, (2) exponential smoothing models, and (3) machine learning techniques.

Autoregressive models, developed by Box et al. [BJR08], comprise the auto-regression and moving average models, which can be combined and extended to the ARIMA model. This ARIMA model is the most general model, while the other models of this category can be described as special cases. Table 3.2 summarizes the presented autoregressive models. Within the process step of model identification, the Autocorrelation Function and its derivative of the Partial Autocorrelation Function can be used to identify the model and its order.

Table 3.2: Summary of Autoregressive Time Series Models

	AR	MA	ARMA	ARIMA	SARIMA
Stationarity	x	x	x	x	x
Trend				x	x
Seasonality					x
Process	Linear Combination	Stochastic	Mixed	Mixed	Mixed
Parameter Estimation	Least-Square-Regression	Exponential Smoothing	Least-Square-Regression	Conditional Sum Square / ML	Conditional Sum Square / ML

The second important basic forecasting approach is the Exponential Smoothing, which is a derivative of the moving average technique (not to be confused with the model of the same name). It can be used for stationary time series (Single Exponential Smoothing), time series with trend (Double Exponential Smoothing), and with trend and seasonal component (Triple Exponential Smoothing). Table 3.3 summarizes the exponential smoothing methods

Table 3.3: Summary of Exponential Smoothing Time Series Models

	SESM	DESM	TESM
Stationarity	x	x	x
Trend		x	x
Seasonality			x
Parameter Estimation	Nonlinear Least Squares, LMA, GMA	Nonlinear Least Squares, LMA, GMA	Nonlinear Least Squares, LMA, GMA

Both, autoregressive and exponential smoothing models are explorative, adaptive methods that are commonly used to describe time series and predict future values. Table 3.4 compares both model classes regarding some facts. However, it can not be determined universally valid in advance which model fits which kind of time series best.

Table 3.4: Comparison of Autoregressive and Exponential Smoothing Models

	Autoregressive Models	Exponential Smoothing
Trend	x	x
Seasonality	x	x
Basis	Autocorrelation	Structural view of the data
Linearity	Linear	Linear/Nonlinear
Trend Basis	Elimination	Estimation
Seasonality Basis	Adding second model	Adding seasonality term
Exogenous Influence	Exogenous term (ARMAX)	Not directly integrated

In addition, several other techniques from the area of machine learning were introduced, namely Support Vector Regression, which is a special form of a regression algorithm, and Bayesian Networks, which calculate a special probability density function.

Furthermore, we presented several methods that can be used to estimate the parameters of a forecast model. This includes the method of least squares and the maximum likelihood function. Finally, we discussed several metrics to measure the accuracy of forecast models for a given time series. All these metrics have different applications and usage scenarios as well as advantages and disadvantages.

4 Related Work in Energy Forecasting

Forecasting of energy data is an important aspect in the energy market in order to enable resource scheduling and balancing of energy demand and supply. The energy demand of customers cannot be planned in a suitable way because they consume energy in a random fashion. In addition, renewable energy sources are influenced by exogenous factors. For this reason they cannot be scheduled reliably. Therefore, forecasting of energy demand and supply is necessary in order to predict the consumption and production. In addition, the scheduling should optimize the energy consumption with respect to the energy price. Since the price may not be known at the time of scheduling, it has to be predicted as well. In this chapter, we examine existing forecasting solutions in the domain of energy demand and supply as well as the prediction of energy prices. The goal is to identify possible forecast model candidates for energy data management systems. In addition, we present existing work in the fields of forecasting in data management systems and distributed forecasting.

4.1 Forecasting Energy Demand

To describe existing forecasting approaches in the energy domain, we have a detailed look at several aspects. First, we review solutions for energy demand forecasting. We also describe the integration of exogenous influences and seasonality to enhance the forecast result. In addition, methods to increase the efficiency of the parameter estimation are presented. Second, we have a look at supply forecasting. Forecasting of energy supply is more complicated because the energy production heavily depends on a large number of uncertain variables like weather and wind speed that also have to be predicted. Third, the forecasting of energy prices is described, which exhibits many abrupt peaks. Special models and configurations are necessary to describe time series in this area.

Energy demand data show multi-seasonal behavior and the demand is influenced by several exogenous factors. Both aspects need to be included into the forecasting of energy demand in order to increase the accuracy of the results. In this section, we describe several existing solutions for energy demand forecasting. We start with the identification of a suitable forecast model that is very important for accurate forecasts. Afterwards, we discuss alternatives to integrate exogenous variables and multi-seasonal behavior. Furthermore, we describe existing solution in the field of parameter estimation.

As presented in Chapter 3, most forecast solutions are either based on auto-regression or on exponential smoothing. We reuse the categorization shown in Figure 3.2 and put the existing energy demand forecast models into this classification as well. The enhanced classification is illustrated in Figure 4.1. In addition, there exist two common classes of forecasting models. First, the *single-equation* models describe the behavioral aspects in a single forecasting model, which tends to be very complex. Second, *multi-equation* models use one model for at least each hour of a day, exactly describing the behavior of this specific timeframe. This model class avoids the modeling of the complex intra-day behavior to ensure simpler equations. Unfortunately, more model parameters have to be estimated. Note the emphasized models (bold font) that we will use for our evaluation of existing techniques in Chapter 5.

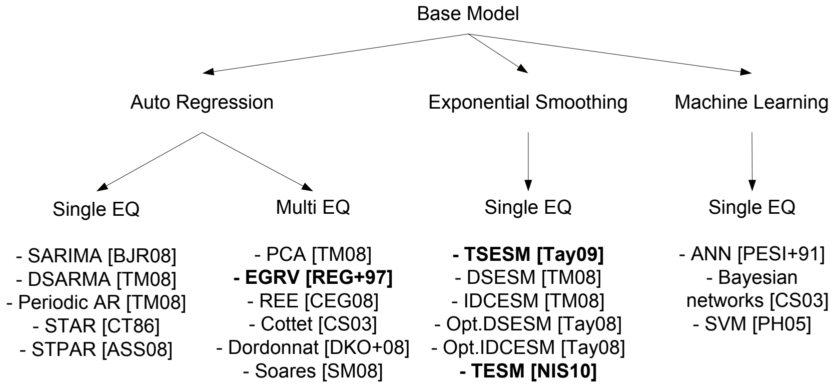


Figure 4.1: Classification of Existing Techniques for Energy Forecasting

Identification of a Forecast Model

The first step in the forecast process is the identification of a forecast model that accounts for the respective data set (compare Chapter 3). We based our review of existing techniques on the comparison of several short-term energy load forecast models conducted by Taylor [TM08]. The evaluation included the following models.

- **Seasonal ARMA Modeling (DSARMA):** The Seasonal ARMA Modeling is a derivate of the standard Seasonal ARIMA as described in Section 3.1.

- **Periodic AR Model:** An issue of the ARMA modeling is that the intra-day cycle of the weekdays is very similar, while it is different for the weekends. ARMA Models cannot directly describe such variations. Therefore, a model that allows parameter changes within the seasons is necessary. This feature is captured by periodic models. The Periodic AR Model chosen for this evaluation is a representative of the ARIMA models [BJR08] that uses auto-regression only.

- **Double Seasonal Exponential Smoothing (DSESM):** This model is derived from exponential smoothing. Taylor chose a solution, which multiplicatively integrates the intra-day and the intra-week cycle.

- **Intra-day Cycle Exponential Smoothing for Double Seasonality (IDCESM):** Similar to the issue of the Seasonal ARMA. the standard Seasonal Exponential Smoothing assumes the same intra-day cycle for all days. Therefore, Taylor suggested an exponential smoothing method that is tailor-made for the intra-day cycle.

- **Principal Component Analysis (PCA):** Besides the single-equation models, a multi-equation model was also included. The PCA maps a multivariate time series to a set of orthogonal variables, while simultaneously reducing the number of dimensions. The PCA variables are linear combinations of the variables from the time series.

The test data consists of 20 weeks of energy data from ten European countries. The results show that Double Seasonal Exponential Smoothing performed best, followed by the PCA model and the seasonal ARMA. It was surprising that both intra-day cycle aware methods did not perform as well as the common methods. Taylor supposed that 20 weeks of training data might not be sufficient. The average MAPE for all forecasting models was below 2,5% which is a good result for energy data forecasting.

Most of the models considered in Taylor's evaluation [TM08] are single-equation models. As stated in the introduction of this chapter, there also exist several multi-equation approaches that use one model for at least each hour of a day. This technique covers the intra-day cycle and it can react faster on certain intra-day events. The first approach considering a multi-equation model was introduced by Ramanathan et al. in [REG+97]. They use one separate linear regression model for each hour of the day. Each model includes several dependencies, which leads to the problem that for the calculation of all 24 hourly models 393 parameters have to be estimated. To solve the estimation in reasonable time, they use the Maximum Likelihood and the Least Square Estimation methods. In addition, they assume that several parameters are the same over several hours and therefore, the number of parameters is reduced. They also present an approach how to adjust the model for a persistent error during the hours. This is done in addition to the persistence of the lagged errors in the equations anyway. Ramanathan achieved an average MAPE between 2.99% and 7.70% which ranks his forecasting approach behind the single-equation models tested by Taylor. However, in the year the model was invented it won a forecasting competition over all other models. Since then the approach was adapted and enhanced by several other people that created multi-equation forecasting models such as Cancelo et al. [CEG08] (REE), Cottet et al. [CS03], Dordonnat et al. [DKO+08] and Soares et. al [SM08].

Besides the approaches based on auto-regression or exponential smoothing, other possibilities exist to forecast energy demand data. For example artificial neural networks (ANNs) were considered in [dSFV08, PESI+91, TdMM06]. The ANN delivered worse results in [TdMM06], which stands in contrast to both other approaches. Da Silva [dSFV08] and Park [PESI+91] considered Artificial Neural Networks as a good solution for forecasts in a short-term horizon because they can capture non-linear interdependencies between load and exogenous variables. In addition, they can react fast on abrupt changes in the behavior of a time series. In [TdMM06], the neural network achieved a mean MAPE between 3% and 4% for a 24-hour-ahead prediction of electrical data from Great Britain. The MAPE for the ANN in [dSFV08] is between 0.5% and 1.9%

for a 6-hours-ahead prediction of Australian data. With other data sets from the USA and Slovakia they produced an average mape of 4.9% (USA) and 1.9% (Slovakia). In [PESI+91], they evaluated an ANN using data from the USA. As well as in [dSFV08] they excluded holidays from their evaluation. The MAPE in their evaluation is between 1.40% and 2.06%. Despite the good accuracy results, there exists several problems with ANN forecast solutions. ANNs tend to be very complex and the training of such models is very time consuming. In addition, the accuracy potential of an ANN highly depends on the selection of suitable input variables that is often done manually [dSFV08]. Potential solutions like the automatic input selection proposed by da Silva et al. [dSFV08] have to be considered to overcome some of the problems. However, for time series that incorporate exogenous variables with a non-linear dependency, ANNs should be considered as a potential solution.

Another interesting forecast approach is introduced by Pai et al. in [PH05], where they utilized a support vector machine for their forecasting. SVMs minimize the upper bound of the generalization error instead of the training error as it is done in traditional forecast models. In addition, solutions found by an SVM are always unique and globally optimal, since they are equivalent to solve linear constrained programming problems. In [PH05], several adoptions of SVMs for forecasting non-stationary time series are shown. They achieve an average MAPE of 1.76%. Therefore, the usage of SVMs for forecasting problems is reasonable. It has to be mentioned that the accuracy highly depends on the number of input data. For example, the average MAPE increases to 3.155% for 25 points of input data.

The identification of a proper model is an important step to forecasting with high accuracy. However, it is also necessary to address the influence of exogenous variables that influence the energy demand.

Inclusion of Exogenous Influences

The energy demand depends on many exogenous influences. A selection of those influences is illustrated in Figure 4.2.

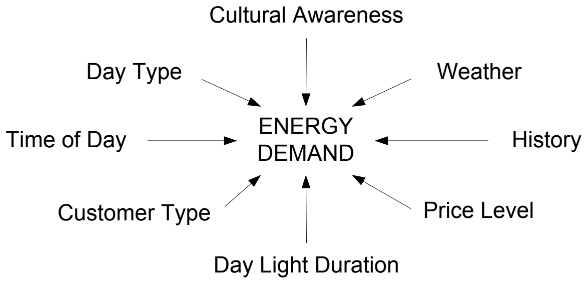


Figure 4.2: Dependence of the Energy Demand on Exogenous Influences

It is necessary to include these dependencies to increase the accuracy of a forecast model. Several integration approaches exist. For multi-equation models, the integration is often accomplished through dummy variables as described by Ramanathan et al. [REG+97]. Dummy Variables are binary variables that indicate the existence of a certain effect. For this reason, they can have the values zero or one. In combination with weight parameters, they are used to integrate dependencies by quantifying their influence on the specific data. Therefore, the dummy variables in this setting describe the concrete influence of a dependency to the load for the specific hour. The influences are divided into several classes. The first class consists of deterministic variables that are reasonably predictable. This includes influences like the day of the week or the month of the year. Influences that are not necessarily predicted correctly fall into the second class. This includes influences like weather or temperature. Also second class influences are non-linear in most cases and have to be modeled accordingly. The third class describes measurements at a specific point in time. This includes, for example, the load of a system. The following example is taken from [REG+97] and shows the model for one hour. The model incorporates several dependencies such as the day of the week, the temperature, the previous load and the past errors.

Example 3. Multi-Equation Forecast Model: *Assume a single forecast model for one hour of a day. The full multi-equation model would therefore consists of 24 such models (or 48 models if we additionally distinguish between weekdays and weekend). In general, a single model can be described as follows:*

$$\begin{aligned} \mathit{HOUR1} = & 1678.9 + 40.6\,\mathit{YEAR} + 32.4\mathit{DEC} + 0.5\mathit{FEB} \\ & + 71.7\mathit{MAR} - 193.9\mathit{MONDAY} + 223.9\mathit{DAYAFTERHOLIDAY} \\ & - 2.86\,\mathit{TEMP} + 0.39\,\mathit{TEMP2} - 7.68\mathit{SEVENDAYAVGMIDTEMP} \\ & - 3.62\,\mathit{YESTERDAYMAXTEMP} + 0.08\mathit{LOAD8A.M.} \\ & + 0.07\mathit{MONxLOAD8} - 0.10\mathit{DAYAFTERHOLIDAYxLOAD8} \\ & + \mathit{PASTERRORS} \end{aligned}$$

Besides the base load (first variable), several other variables are included. Each variable is multiplied with a weight parameter in order to fit it to the training data. Included are variables for the annual trends (YEAR), month of the year (DEC, FEB, MAR), day of the week (MONDAY) and the type of the day (DAYAFTERHOLIDAY). They also include temperatures measured at 1am (TEMP), the square of the temperature (TEMP2), max temperature of the previous day (YESTERDAYMAXTEMO), the seven day average (SEVENDAYAVGMIDTEMP) temperature, and interactions of temperature and monthly binary variables. Finally, they include the load at 8am of the previous day (LOAD8A.M.) and its interactions with Monday (MONxLOAD8) and the day after a holiday (DAYAFTERHOLIDAYxLOAD8). In addition, the forecast errors of the previous days are included in order to adjust for those previous errors.

All dependencies are additively included into the model of the specific hour with each variable weighted differently. For another hour, the same set of variables with different weights and, in some cases, with different values are used. A multiplicative inclusion is also possible. Many other multi-equation approaches like [CEG08] or [CS03] use the approach of dummy variables. In most solutions, dependencies are integrated in a linear model but the relationship of several variables is non-linear, e.g., the dependency between load and temperature. Besides ANNs and SVNs that are able to model non-linear dependencies, there exist special non-linear forecast models. Two examples are the Smooth Transition AR model (STAR) [CT86] and the Smooth Transition Periodic AR model (STPAR) [ASS08], which is the periodic extension of the STAR. The models allow a smooth transition between several sets of conditions with each set being modeled as an autoregressive process. In addition, they define a transition function that determines the data composition of each set. They state that this models the asymmetric correlation between temperature and load best. However, this approach is usable for other variables with asymmetric dependency as well. Other models like [DKO$^+$08] incorporate a special modeling of the non-linear correlation between selected variables only. One special feature of their model is that the annual pattern that is correlated to the temperature is modeled as a non-linear heating function. This function is then included into the multi-equation model in order to describe the dependency between load and temperature.

For single-equation models, several possibilities to integrate exogenous influences exist. For the models introduced by Box and Jenkins [BJR08] a variation called ARMAX was proposed (compare Section 3.1). The ARMAX model allows the inclusion of exogenous variables by additively or multiplicatively adding a term for the exogenous variable to the equation. This can be done for several different dependencies as well. This is similar to the dummy variable approach but with a more sophisticated modeling of the exogenous influence. Other single-equation models like Exponential Smoothing approaches do not directly integrate exogenous influences other than previous values or seasonality. Often the model is specifically adapted to include the effect of exogenous influences such as described by Souza et al. in [SBdM07].

Inclusion of Seasonal Behavior

Time series of energy demand data exhibit multi-seasonal behavior. There exists reoccurring patterns with regard to the intra-day, weekly, and annual cycle. This behavior should be addressed by the forecast model. An open question is the number of seasons that are integrated in the model. We divided the existing solutions regarding the number of seasons they incorporate. This categorization is illustrated in Figure 4.3. Again, we emphasized the models used within our evaluation. This selection was made in order to evaluate a representative model from each major category, namely with single, double, and triple seasonality as well as with single- and multi-equation structure.

Beside the number of seasons that are considered in a forecast model, the more interesting question is how they are incorporated into the model. For single-equation models, a special term or equation is added or multiplied for every season considered by the model. For example, Double Seasonal Exponential Smoothing incorporates one term for the daily season and one term for the weekly season. Both seasons are multiplicatively included into the exponential smoothing model. The model can be enhanced with additional seasons by simply adding another parametric equation. Taylor tested

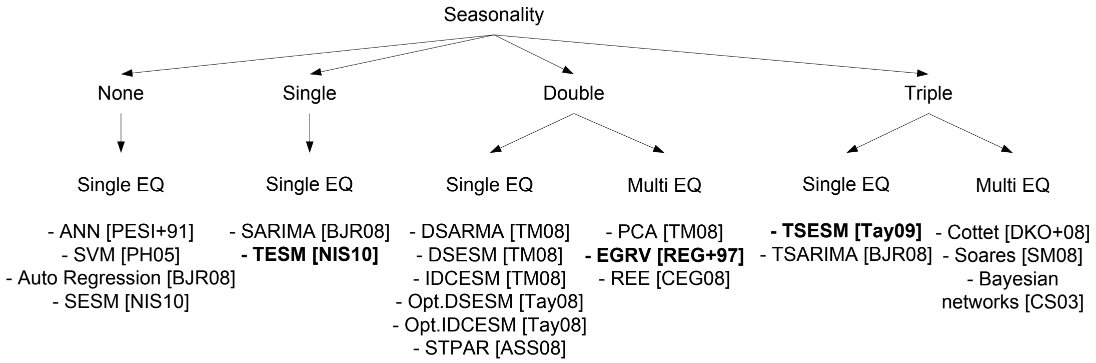


Figure 4.3: Existing Solutions Categorized by the Number of Integrated Seasons

in [Tay09] the integration of a third season into double seasonal models, namely two exponential smoothing methods and one ARMA approach. This added season is the additional intra-year seasonal cycle. The annual season basically describes the annual reoccurring behavior during the meteorological seasons of the year. The evaluation [Tay09] considers three configurations for each forecast model:

1. Double Seasonal Configuration 1: Including intra-day and intra-week cycle
2. Double Seasonal Configuration 2: Including intra-week and intra-year cycle
3. Triple Seasonal Configuration: Including all three seasons

All tested forecast models and configurations achieved a maximum MAPE of less than 2% for lead times of up to 24 hours. The results show that the Triple Seasonal Configuration outperforms both Double Seasonal Configurations starting at a lead time of three hours and above. In addition, the DS Configuration 2 of the Exponential Smoothing showed the worst results, whereas the Configuration 1 of both models and the Configuration 2 of the ARMA model showed results at the same level. The best overall results were achieved by combining the forecasts of the Triple Seasonal Exponential Smoothing and the Triple Seasonal ARMA. As a result, from an accuracy point of view, the consideration of all three seasons is indeed worthwhile. However, using the Triple Seasonal Configuration, the error was decreased by less than 0.5%. Even a decrease of only this magnitude can lead to a huge financial benefit, since one percent increase in forecast error increases the costs by ten million pounds [SM08]. However the incorporation of an additional season also increases the calculation effort of the model. Therefore, it would be necessary to quantify the tradeoff between the increase of accuracy and the increase of the calculation effort.

An approach on efficient integration of multiple seasonality in an additive or multiplicative way was introduced by Gould et al. in [GKO+08]. They divide the seasonal cycles into several subcycles and combine subcycles that are similar. With this technique, they reduce the number of parameters that need to be calculated and therefore, reduce the calculation time. They describe their method for Double Seasonal Exponential Smoothing and a new approach based on a state space model called Multiple Seasonal Process. In addition to the support of multiple seasons, they allow cycle changes during the season rather than only at the end of a season. The approach of dividing one season into several subcycles is similar to the multi-equation approaches.

In multi-equation models such as [DKO+08] or [REG+97], the daily season is eliminated by having a single equation for every hour of a day. Weekly and annual seasons are described by variables for the day type or for the current date. These variables reflect the reoccurring events in the forecast model. For example, an extensive categorization of days with special parameters for every category is presented in [SM08]. The same can be done for annual seasons (e.g., summer and winter) or other exogenous influences. This approach leads to additional parameters that have to be estimated for each hour of the day.

Seasonality is a periodically reoccurring behavior. Therefore, it can be represented by a periodic function as well. A special model called STPAR [ASS08] describes seasonality by a Fourier representation. Let us use an example to illustrate this approach.

Example 4. STPAR Seasonality Representation: *The STPAR model represents the intra-day seasonality as a fourier function in the form of a combination of sin and cosine terms:*

$$\beta_{R,i} = \alpha_{R,i} + \sum_{j=1}^{h} \left\{ \lambda_{R,ij} sin\left(2j\pi\left(\frac{D(t)}{48}\right)\right) + \gamma_{R,ij} cos\left(2j\pi\left(\frac{D(t)}{48}\right)\right)\right\}.$$

A function for the intra-day cycle that quantifies the periodicity is denoted by $D(t)$ (e.g., $D(t) = 1, 2, ..., 48$ for half-hourly load data). The currently used configuration is described by regimes R (a number of different models with different configurations) that is integrated to allow the modeling of non-linear dependencies and a fast switch between different configuration sets.

More periods can be integrated by adding additional Fourier functions to the model. A similar approach has been introduced by Soares et al. in [SM08]. They model the seasonality as waves of sine and cosine. The model is called Two-Level Seasonal Auto Regressive Model (TLSAR) and it can be used for real-time online load forecasting. The model is based on the auto-regression forecast model with an adaptation for periodicity. Certainly, they describe possibilities to adapt their approach to other models. Periodic models have the advantage that they take the variation of the autocorrelation at particular lags into account, e.g., of one half hour; whereas models like seasonal ARIMA do not. In addition, due to the periodic functions (sin, cosine), the number of parameters is reduced compared to non-periodic forecast models. The average MAPE they achieve for a 24 hours forecast is between 2.86% and 3.56%. The integration of seasons as a periodic function is a considerable approach to reduce the number of parameters, while keeping a good accuracy at the same time.

Estimation of Forecast Model Parameters

All variables and dependencies integrated into a forecast model, incorporate additional parameters as well. All parameters in a model need to be estimated, while creating the model. For a large number of parameters, this could take a lot of time. For example, for the model proposed in [CS03], the calculation takes several hours to days. This is a challenge when a lot of updates occur, like in the energy domain, and the forecast model validation is performed during runtime. If a parameter recalculation becomes necessary, the time frame for the calculation should be as short as possible. One solution is the incremental maintenance of the model, which means the reuse of already calculated results. This can reduce the calculation time significantly. However, most forecast models do not support incremental maintenance. Therefore, a sophisticated approach for reducing the number of parameters or calculating the parameters in an efficient way is necessary. Standard methods to estimate the parameters are the maximum likelihood method or the least square regression. However, for an increasing number of parameters, the calculation effort also increases. One method to reduce the calculation time is the simultaneous calculation of several parameters. Dordonnat introduced an adaptation of a multi-equation approach [DKO+08] creating a model that is designed for changing customer behavior and utility production efficiencies. The model is based on the multivariate Gaussian state space framework and incorporates many dependencies similar to the model in [REG+97]. Therefore, the calculation of many parameters is necessary. To reduce the number of parameters, the dependencies between the equation of different hours are taken into account similarly as with the PCA method tested in [TM08]. In addition, they enhanced the multi-equation approach by developing a possibility to simultaneously calculate all necessary equations. This is done using different methods for all components of the model. Particularly, a Kalman Filter supported by the Maximum Likelihood Method with the Maximization Expectation algorithm is used. With their method, they achieved an average MAPE between 1.31% and 1.5% for an one-day-ahead forecast horizon of the French load data. Certainly, the approach [DKO+08] only calculated a model for two hours of a day. A simultaneous calculation for all 24 hours would be necessary for a sophisticated day-ahead forecast. In addition, the correlation test between several hours of a day as performed in [DKO+08] and [TM08] is an interesting approach to reduce the number of parameters. In fact, it is most likely that several parameters do not differ between the different hours of a day, especially, for successive hours. In addition, it can be assumed that some parameters can even be adopted from the same day of the previous week.

Another approach for an efficient calculation of model parameters is described in the forecast solution using an SVM [PH05]. They use a heuristic optimization method called simulated annealing. Simulated annealing is used to find an approximated solution for problems that are too complex to try every possible solution. This gradient method searches for a parameter similar to the cooling of heated metal with the possibility to leave local optima. Each step in the method replaces the current solution with a random solution in the area of the old one. The random solution is chosen with a probability depending on the difference between a value of the analyzed function and a global heating parameter. In their adaptation of this algorithm, they integrate the error calculation to decide whether the found parameters are acceptable or not. The advantage of this approach is that it is possible to take quality of service (QoS) parameters such as the execution time into consideration. This means for example, that a maximal allowed time for the parameter estimation can be defined. Thus, this approach can be classified as a closed-contract anytime algorithm. The best parameters found within this time are taken as the solution. This can be adapted for several forecast models when considering limited resources or time for the computation of the parameter estimates. However, even a small increase of the forecasting error costs a huge amount of money. Therefore, it should be considered to enhance the parameter estimation process to compute in reasonable time without sacrificing accuracy.

Summary

To summarize, the model identification process is the first step in the forecast process. Finding a model that fits the given data best is crucial for an acceptable forecast accuracy. Most of the presented approaches are adapted for data from a special area. For a generic approach, a model is necessary that fits the time series behavior of electricity load for many different countries. Both single-equation models and multi-equation models offer solutions with good forecasting accuracy. Most promising models are the Double/Triple Seasonal Exponential Smoothing [Tay09] for the single-equation model class and the approach from Dordonnat [DKO+08]. In addition, solutions that take a totally different approach like Artificial Neural Networks [dSFV08] or SVMs [PH05] should also be considered as well. The integration of exogenous influences is another important aspect to enhance the forecast accuracy. Several integration methods exist for the inclusion of exogenous variables like dummy variables, exogenous terms or non-linear models. In addition, to the exogenous influence the seasonality of a forecast model is an important factor. It has to be decided how many seasons a model has to incorporate, with the remark that in most existing solutions at least two seasons are integrated. For the integration of a third season the trade-off between the additional effort and benefit has to be evaluated. In addition, there exist some periodic adaptations of available models describing the seasons with sine and cosine waves that reduce the number of parameters included for a season. The most time consuming part of the model calculation is the estimation of the forecast model parameters. Each parameter that is included in a forecast model increases this effort due to the d-dimensional search space when estimating the d parameters. Therefore, parameter reduction methods as proposed for the PCA in [TM08] and for the approach in [DKO+08] should be taken into account. Also methods which reduce the calculation effort or allow the parallel calculation of parameters are necessary. There is a lack of methods that reduce the parameter estimation time without reducing the accuracy.

4.2 Forecasting Energy Supply

The production of renewable energy sources (RES) depends heavily on external factors, and thus, it cannot be planned like for the traditional energy sources. Especially weather conditions strongly influence the production of renewable energy sources. For example, photo-voltaic power plants depend on the cloudiness and the time frame the sun is shining. In contrast, the production in wind energy plants is influenced by the wind speed [JR08, Sán08, WMHM01, Zac03]. Since the different renewable energy sources depend on different influences, different models for each source are necessary. Below, we explain the specific characteristics of renewable energy sources on the example of wind energy but the modeling approach is representative and can also be applied to other sources. The prediction of wind energy is closely related to the prediction of stochastic atmospheric variables like wind speed, wind direction and air density. This is extremely difficult due to the wide variety of spatial and temporal scales of atmospheric motion. Zack et al. divided the forecasting problem into several different subproblems regarding the horizon of the forecast [Zac03]. The categories they found are very-short-term (0 to 6 hours), short-term (6 hours to 3 days) and medium-term (3 to 10 days). Each category has its own particular characteristics. For very short-term forecasts, the observation and prediction of changes of the atmospheric features is necessary. Those forecasts heavily depend on real-time data from the wind plant instead of the pure predictions of the conditions. In contrast, short-term forecasts are linked to the regional prediction of the influencing variables, and measurements at the power plant have much smaller effect. For medium range forecasts the analysis of continental and global atmospheric systems is necessary. Measurements at the power plant side are almost neglectable. Each horizon has different pieces of information to base the forecast on. In addition, the information necessary to forecasts a longer horizon are more and more uncertain and are also based on predictions and calculations.

There are two common approaches described in the literature to conduct the forecast of energy supply from renewable energy sources. The first one is an ensemble approach as described by Sanchez et al. in [Sán08] and by Zack et al. in [Zac03]. Several forecasts are combined through a linear combination to create a result better than individual forecasts. The linear combination has to be conducted very carefully to ensure that the combined forecast is better than or at least equal to the best individual forecast. It should not be possible to create a result that is inferior to a single forecast. The combination procedure should be very flexible to ensure the following two aspects. First, learning and adaptation happens online without explicit training and user intervention. Second, it is possible that the relative accuracy varies with time and forecast horizon, which means that the combination has to be modeled accordingly [Sán08]. Therefore, the following two step approach is suggested in [Sán08]:

1. **Combination of Improvement:** This is a regression method that finds the best constrained linear combination for a set of forecasts.

2. **Combination of Adaptation:** It is a dynamic model selection method that

assigns the highest weight to the best available forecast model and integrates prediction errors with an exponential weight.

In the first step, all available forecasts are combined using the Combination of Improvement method. In the second step, this combination together with all individual forecasts is treated as a new problem and the Combination of Adaptation method is used. While the first step is conducted to find the best linear combination, the second step ensures that the combination is as least as good as the best individual forecast. The parameters and weights of the linear combinations are estimated using exponential weighted moving average, recursive least square or covariance based procedures [Sán08].

The second approach is based on evolutionary forecast methods that rapidly adapt to new situations [JR08, WMHM01]. This approach is especially useful for very-short-term forecasts because unpredictable changes of the weather cause rapid influence in the production of renewable energy sources. In [WMHM01], a system is introduced that bases its forecasts on the prediction of the probability distribution of the power generated. They recommend the use of Artificial Neural Networks (ANN) for this task. ANNs are highly adaptive and are often used in situations, where abrupt changes can occur. They combine the approach with the QoS specifications in order to ensure a limitation of the execution time. This means that a usable result is available after a given time, which is than incrementally enhanced to find the optimal solution. Certainly, they state the time needed for the initialization of the ANN with one to several hours. However, after the first initialization, the adaptation is possible in much less time [WMHM01]. They achieve an average MAPE of 20% with their approach due to the stochastic nature and heavy weather influence. Therefore, they also suggest reserves and energy storages to compensate for the high MAPE. A similar approach is introduced in [JR08] with the difference that they propose the usage of two models. The first model is the physical model considering predicted weather data, historic power data, etc. The second model is the mathematical model that uses statistical methods like ANNs to model the correlation between the weather data and the energy production. In addition, they integrate data from several other windfarms appart from the one the forecast is created for. Therefore, they have additional space/time information available helping to predict the weather conditions in the near future. However, because of the large number of data sources, an algorithm for selecting relevant variables is necessary.

To summarize, accurate forecasting for the prediction of the power supply for renewable energy sources, is much more complicated than forecasting energy demand because the sources highly depend on several stochastic processes like weather. Both presented approaches are a good starting point for further investigation. They address the specifics of renewable energy sources, e.g. rapid changes, and dependency on exogenous influence, in a very convenient way.

4.3 Forecasting Energy Prices

The energy market is often organized as an auction market for energy, which makes it difficult to predict the price [KB08, WM08]. In addition, heavy abrupt peaks occur that are hard to predict beforehand [Bun00, KB08, WM08]. Karakatsani identified five characteristics of energy prices [KB08]: (1) the instantaneous nature of commodity, (2) the shape of supply function, which is steeply increasing, discontinuous, and convex, (3) the exercise of market power coming from oligopolistic market structures, agents asymmetry, negligible demand elasticity of prices, (4) the complex market design, and (5) the substantial agent learning due to highly repeated auctions, frequent regulatory interventions, and market structure changes.

The models used for the price forecasting have to capture the impact of economic, technical, strategic and risk factors on prices and the dynamic of these effects over time. Altogether, spot price forecasting is non-trivial [Bun00, KB08, WM08]. Standard stochastic models have the problem that they are only partially able to describe abrupt, fast reverting peaks. Most of these models are limited to autoregressive effects, weather influence and price responds to demand conditions. Indeed, only models that incorporate exogenous variables are in any way suitable. However, models that can in addition describe plant dynamics, risk measures, market design effects, agent learning, strategic behavior, etc., are necessary for an accurate prediction [Bun00, KB08]. A complete list of necessary exogenous variables is provided in [KB08].

Karakatsani proposes the use of one of the following three forecast models [KB08]. First, linear regression based on market factors captures an average price over the sample period. Second, the Time-Varying Parameter (TVP) regression model allows a continuous, adaptive price structure. Third, the Markov Regime-Switching regression model allows discontinuities in pricing due to temporal irregularities. The models proposed in [KB08], which represent market fundamentals and can adapt to time-varying effects, performed best in the evaluation namely the TVP and the Regime-Switch regression. They achieved an average MAPE of four till five percent. Weron et al. introduced another approach integrating semi-parametric models in his evaluation [WM08]. Most of those models are enhancements of auto-regression and ARMA models. They are enhanced with e.g., spike preprocessing, mean reverting jump diffusion and semi-parametric AR. Weron found that these semi-parametric models perform very well under diverse market conditions. Note that he used only data from the Nord Pool (Norway, Denmark, Estonia, Sweden, Finland) market and California for his evaluation that might be very different from the price data of other countries. However, the statement that models, which incorporate system load as exogenous variables, perform better than pure price models, is true for data from other countries as well. The proposed models achieved an average weekly-weighted mean absolute error (WMAE) for the Californian data from 1999 to 2000 of between 12.96 and 15.39. For the Nord Pool data the average WMAE was between 4.04 and 4.99.

Bunn [Bun00] states that the segmentation of forecasting variables in separate models can lead to better forecasting results. This is similar to the multi-equation approach for the energy demand forecasting. In fact, it is also proposed to use one model for each hour in order to compensate for intra-day effects [Bun00, WM08]. This leads to simpler models instead of one single, complex model. The trade-off is the same as with the multi-equation models for demand forecasting, which means the involvement of more forecasting variables and, therefore, higher computational complexity. Bunn also suggests that the weighted combination of forecasts, similar to the forecasting of energy supply for renewable energy sources, enhances the result.

The complex nature of the market, much exogenous influence and abrupt peaks make the energy prices hard to predict. Some solutions exist that try to solve the forecasting problem. The solutions presented in [KB08] achieved an average MAPE between four and five percent which is very good for the problematic nature of energy price forecasts.

4.4 System Architecture

In addition to the field of forecasting domain-specific data, we now investigate the support for forecasting in data management systems from a system architecture perspective.

Forecasting in Data Management Systems

In contrast to the mathematical perspective that mainly focuses on the forecast accuracy, the integration of forecast functionality into data management systems additionally takes the efficiency of forecast computation into account. In general, there are two relevant system categories: data base management systems (DBMS) and data stream management systems (DSMS), where forecast query processing has already been integrated.

Forecasting in DBMS. Forecasting has already been successfully integrated into DBMS. For example, within the Fa system [DB07], declarative forecast queries can be issued, where execution plans with specific forecast operators are automatically generated. In addition, they also proposed an algorithm for plan adaptation in the presence of continuous queries. Furthermore, the skip-list approach for efficient forecast query processing [GZ08] proposes an I/O-conscious skip list data structure for very large time series in order to enable the determination of a suitable history length for model building. Unfortunately, both approaches do not address the efficient model maintenance in the presence of high update rates.

Aside these research prototypes, forecasting has also been integrated into commercial DBMS. Examples are the Oracle OLAP DML [Ora08] and the Microsoft SQL Server data mining extension [Mic08, Mic07]. In general, built-in and custom forecast algorithms can be used as user-defined functions or using other extension points such as the Microsoft integration services. However, these commercial products require the explicit creation and usage of forecast models. Thus, the user has to specify the forecast model (model identification) and must trigger re-estimation manually (model evaluation and model adaptation). In conclusion, these products do not allow for a seamless integration of the whole forecast process (from model identification to model adaptation), which would require the transparent creation and usage of forecast models that should be invisible to a user.

Forecasting in DSMS. In contrast to the area of DBMS, forecasting is more often applied for monitoring sensor networks. Hence, DSMS that process continuous queries over data streams are an important system category. As already mentioned, the Fa system [DB07] supports continuous queries by forecasting sliding data windows. Furthermore, Akdere et al. addressed continuous prediction queries over streaming data [ACU10] by using recursive execution trees for dynamic Bayesian networks. One important aspect in sensor networks is pre-aggregation [MNG05] in order to reduce the amount of exchanged data. In this context, forecasting is used for accurate interpolation of missing time series data [LGS06, CDH10]. Again, most of these approaches do not focus on the problem of when and how to reestimate the parameters of the created forecast models efficiently.

Chiky et al. introduced two interpolation algorithms for asynchronous electric power consumption time series [CDH10] in a distributed system architecture. Especially in distributed environments, time series in most cases are observed at different timestamps in combination with information about the total amount of used power consumption. The requirement of analysis motivates the use of interpolation to calculate the missing values. In their approach, Chiky et al. assume that the integral between two data points is known, such that it can be used for interpolation calculation. The first algorithm is the naïve interpolation approach that uses the history of slope values to estimate the values between the two given data points. The second approach is the stochastic approach that uses a stochastic process for the calculation. This approach provides a confidence interval in addition to the interpolated values. Due to the design for distributed environments, this interpolation approach might be also applicable distributed energy data management of energy demand and supply.

Apart from these approaches that integrate forecasting into data management systems, the new application requirements of *advanced analytics* (clustering, classification, forecasting, and association rules) in general and *operational BI (Business Intelligence)* [O'C08, DCSW09, WK10] will lead to significant changes of traditional data management systems in the future. One of these changes is the integration of algorithms for advanced analytics (so-called deep analytics) into the execution environment of the data management systems [CDD+09]. For example, within the IBM research project *eXtreme Analytics Platform (XAP)*, the prototype Ricardo enables the decomposition of complex analysis algorithms in order to support the hybrid execution with both the statistical computing software *R* and the Map Reduce implementation *Hadoop* [DSB+10]. In consequence of this trend, also forecasting will be integrated in more data management systems.

Distributed Forecasting

Unfortunately, there are no distributed forecasting approaches available in the literature that can be directly applied for a distributed energy management system. However, in this subsection, we present some interesting partial solutions in this context.

One approach presented by Brabec et al is a modular model called nonlinear mixed effects model (NLME) [BKPM08]. The model is created for the deregulated gas market and can be implemented directly at the customer level. However, they assume a simple aggregation of forecasts on the next higher level, which leads to high communication and calculation efforts. The NLME model contains several structural parts. Each of these parts has a real world interpretation and can be adjusted separately. Therefore, the model can be adjusted to different characteristics of different customers at different locations. Furthermore, the structured nature is useful for model checking and evaluation.

Besides the regional differences, the customer behavior is very different, the flexibility of the model is very useful to create several models with respect to the behavioral specifics. In addition to the fact that the individual estimates are only aggregated on the next higher level, the accuracy of the proposed NLME model is limited in comparison to ARIMAX and ARX. The NLME is good in differencing between several customers, while other time series approaches can react faster on sudden and abrupt changes [BKPM08].

Another distribution approach is forecasting using a distributed sensor network. For example, in wind parks, every wind turbine has several sensors producing partially over hundred measurements that have to be incorporated into forecasting [Zac03]. Unfortunately, forecasting for each wind turbine is not done at the wind turbine but in a central unit. This might be a difference to a distributed energy management system solution. However, the technology of incorporating many values from several units is interesting for such systems as well. Challa describes one such approach in [CCC+05]. They state that Artificial Neural Networks (ANN) are quite useful in this scenario because they handle non-linear relationships between load and exogenous factors directly from historical data without explicitly selecting a model. Due to the argument that single-equation models do not consider real-time information in between their intervals, they developed a state space multi-equation parametric regression model using Vector Auto Regression (VAR). This model allows the integration of real-time measurement data from sensor networks and has a resolution of half an hour. For this reason, sensor data can be updated every 30 min. In order to integrate real-time data, they incorporated different models into their process and calculate the forecast using the Interacting Multiple Model (IMM) estimation algorithm [BBS88]. The IMM is used for recursive state updates and model selections. They measured the MAPE with and without the use of real-time data from distributed sensor networks. Using the real time data, they reduced the average MAPE by around 0.4%. However, the integration of real-time data also means the ad-hoc calculation and the adaptation of forecasting models, respectively. This might not always be possible. In addition, they do not address how to efficiently integrate information from many sources that might have, e.g., delays in providing data.

Chen and Guo proposed a Power Management Decision Support System (DSS) that is based on a multi-layer architecture with several forecast models [CG09]. These models are quadratic exponential smoothing, curve fitting, multiple linear regression models and a gray-box model. All models are made available via a web service, so that they can be called dynamically. To avoid the use of a model that might not fit the data, they perform a regression test and ensure that the relative forecast error of each model does not exceed ten percent. Afterwards, the forecast manager of their system decides which model to choose for a specific forecast. The advantages they see in the use of Web service technologies is that many forecasts can work in parallel and that the addition of new forecasting technologies is quite easy. In addition, they state that one major advantage of their system is to use the same forecast model algorithm to forecast different time series, this should reduce the coupling between programs and its running state. The idea of decoupling the forecasts from the program and providing the functionality as Web services is interesting for distributed energy data management as well.

To summarize, there is no distributed forecast approach that can be used to create a distributed energy data management system. However, the approaches described in this subsection give interesting insights into specific subareas of distributed forecasting and some results can be used as a starting point for future investigation; e.g., the usage of a model that can express different locations and customers as proposed in [BKPM08], and the integration of real time data as proposed in [Zac03].

4.5 Summary and Discussion

In this chapter, we presented the state-of-the-art of forecasting energy demand and supply. We considered solutions of forecasting domain-specific data namely energy demand forecasting, forecasting of renewable energy supply and energy price forecasting. In addition, we analyzed existing solutions for the integration of forecasting in data management systems and in the area of distributed forecasting. In general, there exist a lot of solutions for the forecasting of energy demand. Besides the identification of a suitable forecast model, the integration of multi-seasonality and exogenous influences is a crucial task in order to ensure high accuracy. However, the integration of a large number of factors leads to the requirement of estimating a large number of forecast parameters. This can increase the calculation effort for the forecast model. Several approaches were presented that address this issue and tried to reduce calculation time.

The forecasting of energy supply for renewable energy source is complicated because it depends on several exogenous influences that are very hard to predict (e.g. weather, temperature). The existing approaches deliver interesting solutions regarding different aspects. To use the presented approaches, they have to be evaluated using specific data and market specific energy configurations. In addition, it is necessary to care about the asymmetry of energy demand and supply, which means that the supply, especially from renewable energy sources, is not directly coupled to the behavior of the energy demand. This has to be addressed to allow balancing and scheduling of demand and supply. Energy price forecasting is similar to the forecasting of renewable energy sources. It depends on market specifics and the certain behavior of the actors in the market. The prediction is therefore, only possible by incorporation of many exogenous information sources. In addition, the used model has to be able to predict abrupt peaks.

In conclusion, the inherent hierarchical distributed forecasting system, required for future electricity grids, poses several challenges. No distributed forecasting solution exists for the specific system architecture that is required by future energy data management systems. Presented solutions that integrate distribution aspects do only solve several subaspects or describe systems with other requirements. In the following, we return to the described specific data characteristics of energy demand and supply data and evaluate several forecast models presented in this chapter (see Chapter 5). Finally, we reveal research challenges for further investigation in Chapter 6.

5 Experimental Evaluation

Based on the general description of the background of energy data characteristics and time series forecasting as well as the detailed classification of energy forecast models, in this chapter, we compare these techniques in a detailed experimental evaluation. In particular, we use three different real-world data sets for our comparison and evaluate the accuracy of existing forecast models, where we use one general-purpose technique and one technique from each classification group of energy demand forecast models (single-equation, multi-equation).

First, we describe the experimental setting, which includes the used forecast models, the different forecast time horizons as well as the used error metrics. Second, we show example forecasts and explain the characteristics of the used forecast models. Third, we compare the different models, using different time horizons and different error metrics.

5.1 Experimental Setting

In order to allow for repeatability of experimental results, we used the publicly available data set A in order to compare the accuracy of existing forecast methods. Although data set C is also publicly available, its aggregation level is too coarse-grained such that all methods would achieved a fairly high accuracy.

We compare the following three forecast models, where we choose a general-purpose forecast model, and one tailor-made energy demand forecast models from each of the categories of single-equation and multi-equation models. As the evaluation environment, we used the statistical computing and graphics software R [R10].

- *Triple Exponential Smoothing* (TESM): Due to its robustness, the Triple Exponential Smoothing [NIS10] is a widely used general-purpose forecast model and thus, we included it in our evaluation. Furthermore, it is a representative model for the classification categories of single-equation and single seasonality. We used the standard `HoltWinters` function in R, where the parameters (α, β, and γ) are estimated automatically using the `optim` function.

- *Triple Seasonality HWT Exponential Smoothing* (HWT)[1]: As a representative of single-equation forecast models with triple seasonality, we used the Triple Seasonality Holt Winters Exponential Smoothing (HWT) [Tay09]. We implemented the described method in R. For forecast horizons greater than 48 half-hour intervals

[1] Within the classification of forecast models (see Chapter 4), we referred to it as Triple Seasonality Exponential Smoothing Model (TSESM). However, in order to ensure a clear separation from TESM, in this chapter, we use HWT as the abbreviation.

(one day) we continued the seasonal indices with the forecast values. Similar to this, also the level was continued with forecast values for any horizon greater one. Furthermore, we used the `optim` function from the R `stats` package for parameter estimation (optimizer method: L-BFGS-B [BLNZ95], error function: MSE for one-step-ahead forecast).

- *Engle, Granger, Ramanathan, and Vahid-Arraghi Model* (EGRV): In contrast to HWT, we used the Engle, Granger, Ramanathan, and Vahid-Arraghi (EGRV) model [REG+97] as a representative for multi-equation models but also with triple seasonality. We implemented the described method in R, but changed the algorithm slightly. First, we used 48 models (one for each half hour due to different data granularity) instead of 24 hour models for weekdays and 24 hour models for weekends. Furthermore, we excluded several variables (e.g., all weather parameters, after holiday load, year inverse, no variable elimination). We estimated the model parameters with generalized least squares (R package `gdm`).

We trained instances of these models with training data from the beginning of 2002 to the end of 2008 from data set A. Forecasting was realized for the whole year 2009 in order to evaluate the trained models. In order to reflect the different requirements of energy demand forecasting, we used the following four different time horizons during this evaluation:

- *Very-Short-Term:* The very short time horizon is the one-step-ahead forecast, which is 30 minfor data set A. There, over the whole year 2009 the model parameters where not updated but the model was adjusted by the real values.

- *Short-Term:* Subsequently, the short time horizon uses 48-steps-ahead forecasts, which is equivalent to one day for data set A. In order to evaluate the models, we used sliding window semantics and the arithmetic mean over all forecasts (for each real value, we compare 48 forecast values).

- *Mid-Term:* The middle time horizon uses a week as the forecast horizon which are $48 \cdot 7 = 336$ steps. Similarly as for the short-term forecast, sliding window semantics were used in order to evaluated the accuracy.

- *Long-Term:* Finally, the long term time horizon is the one year forecast, which are $48 \cdot 365 = 17{,}520$ steps of data set B. Due to evaluation of year 2009, we do not use sliding window semantics, but a direct comparison of forecast and real values without the adaptation of real values.

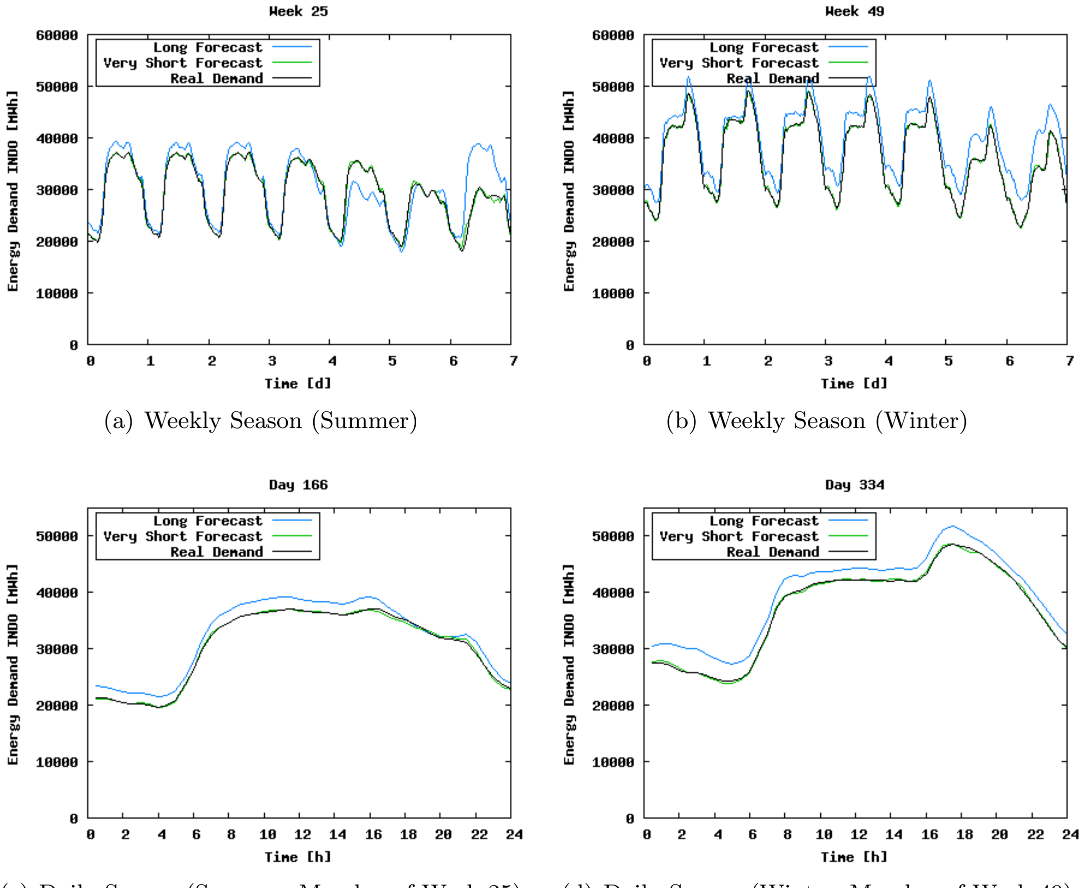
For all of these forecast horizons, we did not re-estimate the parameters of the model during evaluation but incrementally adjusted the actual values.

According to the description of different error measures (see Section 3.5) base on the survey by Hyndman [Hyn06], we used one error metric of each category. In detail, we use the Mean Absolute Error (MAE) from the category of absolute errors, the Mean Square of the Error (MSE) from the category of squared errors, the Symmetric Mean

Absolute Percentage Error (SMAPE) from the category of percentage errors, and finally, the Mean Absolute Scaled Error (MASE) from the category of scale-free errors. In the following, we compare the mentioned forecast models using these time horizons and error metrics.

5.2 Forecasting with TESM

As already described, for triple exponential smoothing (TESM), we used the standard `HoltWinters` function from R that automatically estimates the parameters with the `optim` function. The resulting parameter values differ slightly for the different seasonal indices (48: $\alpha = 1.0$, $\beta = 0.0$, and $\gamma = 0.0$; 336: $\alpha = 1.0$, $\beta = 0.0$, and $\gamma = 0.0$; 17,520: $\alpha = 1.0$, $\beta = 0.0$, and $\gamma = 1.0$).



(a) Weekly Season (Summer)

(b) Weekly Season (Winter)

(c) Daily Season (Summer, Monday of Week 25)

(d) Daily Season (Winter, Monday of Week 49)

Figure 5.1: Data Set A – Triple Exponential Smoothing (TESM)

Figure 5.1 illustrates example forecast results when using TESM. There, we restricted ourself to the very short and the long time horizon because for both time horizons no sliding window semantics are required (which cannot be adequately visualized). For the weekly trend, we observe that the one-step-ahead forecast (very short) is very accurate for both a typical summer week (see Figure 5.1(a) for week 25) as well as for a typical winter week (see Figure 5.1(b) for week 49). In contrast, the long term forecast (forecast from the point of 31$^{\text{st}}$ December 2008) shows larger errors, with a positive bias. However, even this long forecast exhibits the typical shape of evaluated demand data. Note that the error for week 25 is smaller than for week 49 because of the longer forecast horizon for week 49. Figures 5.1(c) and 5.1(d) illustrate the detailed results for the Monday of week 25 and week 49, respectively.

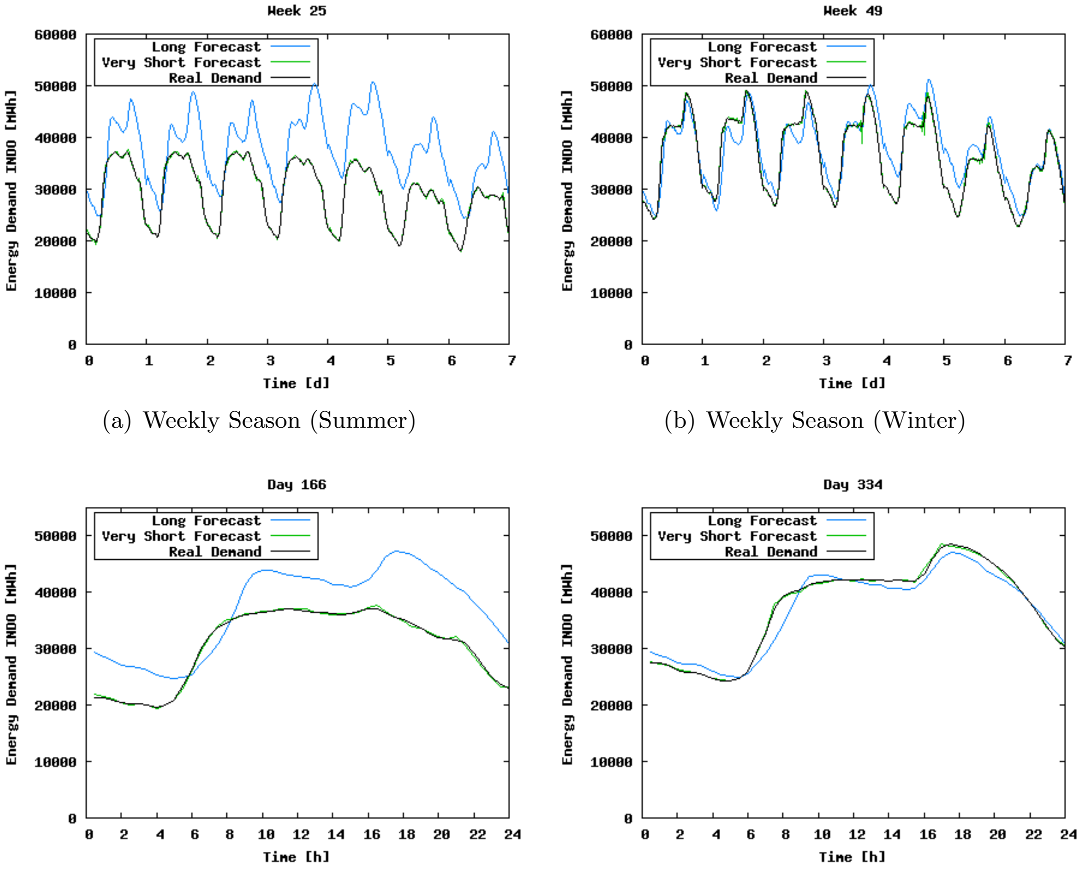
As a result, the error gets worse with increasing forecast horizon. Further, the results get better with increasing number of seasonal indices for all horizons, where we obtained the best results for index 17,520 (one year). The rationale for this is the small change from 2008 to 2009. Thus, it is advantages to have a dependence to the last year rather than to a shorter time period. If we, for example, use a daily season, the model performance suffers from the significant differences between Friday and Saturday and between Sunday and Monday. The mis-predicted Friday and Sunday in Figure 5.1(a) are caused by a leap year in 2008, which led to wrong seasonal indices. However, until winter (see Figure 5.1(b)), the model adjusted these values.

5.3 Forecasting with HWT

In contrast to TESM, the triple seasonality Holt Winters Triple Exponential Smoothing (HWT), is a single-equation forecast model tailor-made for energy demand forecasting. We implemented this method using R as well as the `optim` package for parameter estimation. It is important to note that this resulted in other parameter values ($\lambda = 0.34$, $\delta = 0.27$, $\omega = 0.55$, $\alpha = 0.35$, $\theta = 0.95$) as described by the original paper (because there, they used less training data). Comparing both parameter combinations lead to the result, that the parameters of the original paper performed slightly better for the very-short-term forecast, while the automatically estimated parameters performed better for short, mid and long forecast horizons.

Figure 5.2 shows our evaluation results of the HWT forecast model on the same example data as for TESM. Similar to the TESM evaluation, we restricted the time horizon to very-short and long. We also observe very accurate very-short-term forecast which are slightly worse than for TESM. However, the very-short-term forecasts obtain several peaks, which are lower for summer weeks (see Figure 5.2(a)) as for winter weeks (see Figure 5.2(b)). Most importantly, we observe a fully different long-term accuracy for summer and winter weeks. While in week 25 the error is significant, the model performs well in week 49. Furthermore, the shape of the long-term forecasts is similar to a winter day for both summer and winter days (see Figures 5.2(c) and 5.2(d)).

While the short-term forecast adapt fairly well, the long-term forecast behavior is typical for this model because we start the forecast in Winter 2008. First, weekly and daily indices take only winter days into account. Then, the indices depend on own forecast values with a small influence of the annual index. For this reason, there is a



(a) Weekly Season (Summer)

(b) Weekly Season (Winter)

(c) Daily Season (Summer, Monday of Week 25)

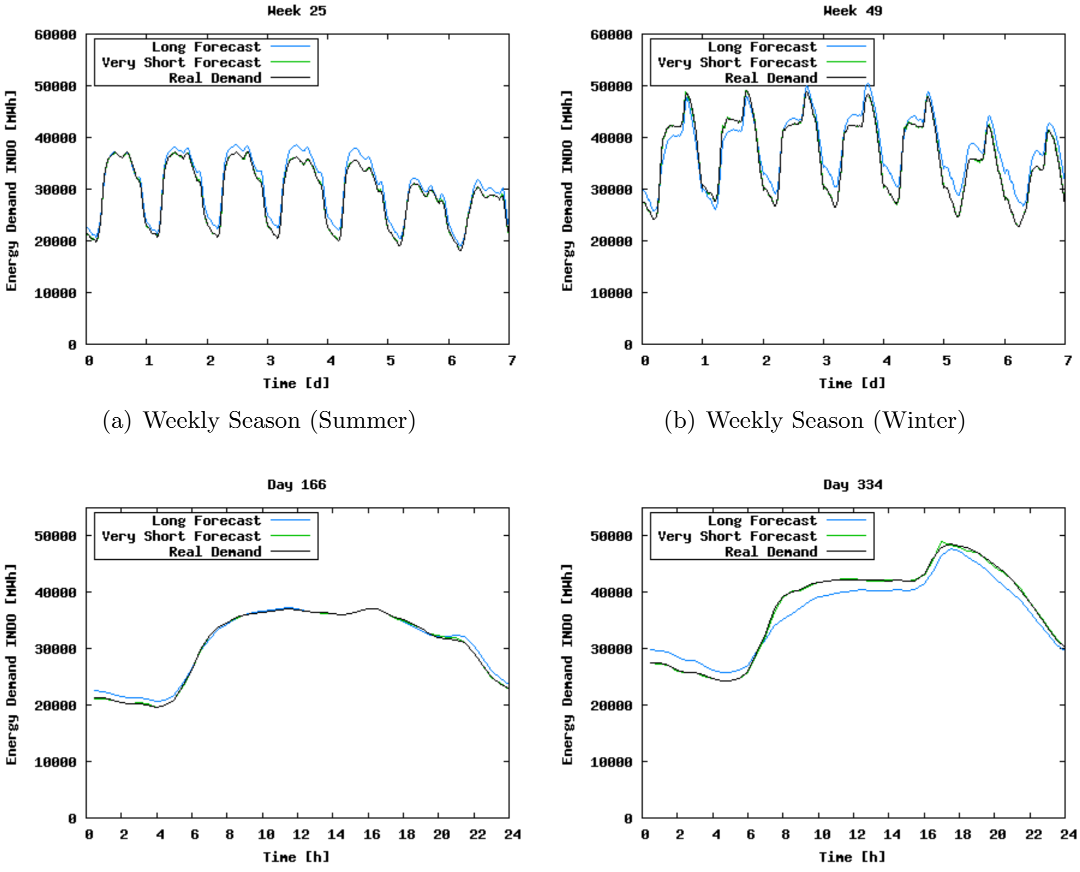
(d) Daily Season (Winter, Monday of Week 49)

Figure 5.2: Data Set A – Triple Seasonality Holt Winters TESM (HWT)

daily seasonal profile of a winter day throughout the whole year. Additional tests with a higher parameter for the annual season and smaller daily and weekly parameters (0.01) lead to the result that the overall shape of summer day was much better represented. However, with this parameter combination, the overall error measures were worse. In general, the error gets worse with increasing forecast horizon.

5.4 Forecasting with EGRV

Similar to HWT, also the hour by hour Engle, Granger, Ramanathan, and Vahid-Arraghi (EGRV) model is a tailor-made forecast model for the energy domain but it relies on a multi-equation model. It uses a model for each time interval of a day in order to reflect the daily season, based on the assumption that the same devices are used during almost the same time every day. Due to the data set resolution of 30 min, we used 48 specialized models. We implemented this EGRV model in R and used the generalized least squares



(a) Weekly Season (Summer)

(b) Weekly Season (Winter)

(c) Daily Season (Summer, Monday of Week 25)

(d) Daily Season (Winter, Monday of Week 49)

Figure 5.3: Data Set A – Multi Equation (EGRV)

(package glm) for parameter estimation of all specialized models. We are also using only very short and long-term forecast. Further, the model update (adjustment of values) uses the forecast as lags as well as the load at 8 am instead of the real values, while all other parameters are constant.

In general, the EGRV model performed significantly better than the two other models as shown in Figure 5.3. Similar to TESM and HWT the very short-term forecast are very accurate for summer weeks (see Figure 5.3(a)) as well as for winter weeks (see Figure 5.3(b)). However, in contrast to the two other evaluated models, EGRV achieves also very accurate long-term forecast. There, the long-term forecast of a summer day (see Figure 5.3(c)) is even better than for a winter day (see Figure 5.3(d)), which is mainly caused by the longer forecast horizon from our starting point of 31st December 2008. In contrast to HWT, the typical summer and winter profile are forecasted as well.

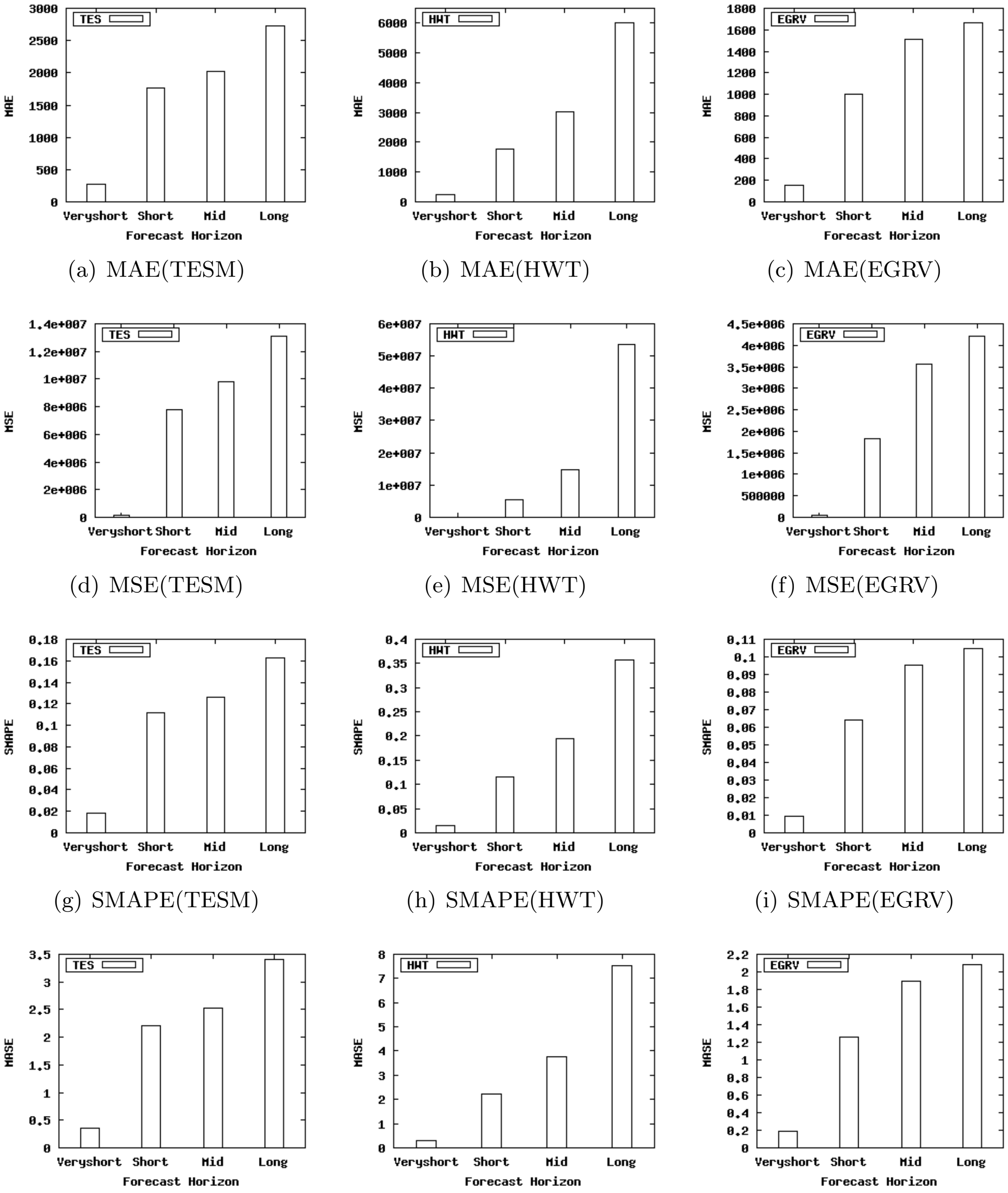
As a result, the EGRV achieved the best results for all horizons. As expected, the error gets worse with increasing horizon. The significant better results on long-term forecast are reasoned by the dependence on many constant (deterministic) parameters such as the day of the week or the month of the year. It is important to note that we did not use any weather parameters, the after holiday load, and the year inverse as well as that we did no variable elimination as stated in the original paper. In conclusion, although the EGRV model was already the best performing model, its accuracy can be further increased by including additional context knowledge.

5.5 Comparison of Forecast Models

So far, we have only shown example forecasts for the time horizons of very short-term and long-term and did not quantify the actual error measures. In this section, we provide overall comparison results of the accuracy of the three forecast models TESM, HWT, and EGRV, using all four time horizons and all four error metrics.

Figure 5.4 quantifies these errors for the different methods, time horizons, and error metrics. Essentially, the EGRV model significantly outperformed the TESM and the HWT model. We observe that this is true for all error metrics and all time horizons. The HWT method was the second best method for the very-short-time horizon, while it performed worst for all other time horizons. This is reasoned by the optimization for one-step-ahead forecasts (very short). Furthermore, it is important to note how the errors increase with increasing time horizon. Using EGRV or TESM, we observe that the measured errors increase logarithmically with increasing forecast horizon for all metrics, while for HWT, the errors increase linearly, or super-linearly with increasing time horizon.

In conclusion, all three methods achieved a high accuracy for very-short-term forecasts. However, only TESM and EGRV did show robustness in terms of an increasing time horizon. Finally, the multi-equation method consistently led to a significantly better accuracy than the other two methods.



(a) MAE(TESM)

(b) MAE(HWT)

(c) MAE(EGRV)

(d) MSE(TESM)

(e) MSE(HWT)

(f) MSE(EGRV)

(g) SMAPE(TESM)

(h) SMAPE(HWT)

(i) SMAPE(EGRV)

(j) MASE(TESM)

(k) MASE(HWT)

(l) MASE(EGRV)

Figure 5.4: Data Set A – Accuracy Comparison

6 Characteristics and Challenges

When comparing the data characteristics of different real-world data sets, we can draw some major findings on characteristics of energy data. In this chapter, we reveal these data characteristics as well as specific characteristics of energy forecasting influenced by the overall process of scheduling energy demand and supply. Putting these characteristics, as well as the results about existing techniques together, we finally discuss research challenges and future work.

6.1 Characteristics

In conclusion of our analysis of energy demand and supply data, the following fundamental characteristics can be derived:

- *Multi-Seasonality:* The behavior of energy demand exhibits multiple significant seasons. First, there is a typical daily season, which shape changes according to winter and summer days. Second, there is a weekly season, where we observe lower demands at the weekend. Third, there is a long-term annual season, caused by exogenous influence (by the weather). In addition to this multi-seasonality, special days such as holidays also have major influence and thus, must be taken into account. As a result, both short term seasons (day, week) as well as special days must be taken into account, while the long term annual season could be used in order to incrementally enhance the results.

- *Exogenous Influence:* Both energy demand and renewable energy supply strongly depend on exogenous influence. At the demand side, for example, the current weather and weather forecasts (including temperature, rainfall, daytime running light) must be taken into account. Further, at the supply side, weather and additional aspects such as the availability of solar panels or windmills need to be taken into account.

- *Aggregation-Level-Dependent Predictability:* When comparing the behavior of energy demand of single households against regionally aggregated behavior, we observe that the lower the aggregation level, the lower the accuracy of forecasts because of higher diversity and often unpredictable behavior. This is true for all aggregation dimensions such as time, customer/producer, and region. Thus, increasing the aggregation-level allows for more robust forecasts (similar to the law of large numbers). However, if the aggregation level is too high many effects are hidden and cannot be precisely estimated. As a result, forecasting energy demand and supply should be designed with aggregation-awareness.

In addition to these characteristics that are mainly imposed by the energy data itself, there are also several characteristics that are created by the process of balancing energy demand and supply within the energy market. These characteristics are:

- *Asymmetry of Energy Demand and Supply Forecasts:* Forecasting consumption and production is not symmetrical. On the one hand, the energy demand forecast (energy consumption) is used 'as is' for planning the production. On the other hand, we forecast energy supply of renewable sources and use this forecast together with the demand forecast in planning the production of other energy sources. Estimated mismatch can be used to determine the energy price of this day or to calculate so-called demand responses [CEC03].

- *Multivariate Forecasts or Multiple Forecasts:* Based on the specific characteristic of request-based energy demand and supply, the total energy consumption and production must be separated into request and remaining consumption. Thus, both the total consumption as well as the consumption by requests must be predicted. As a result, a multivariate forecast model or at least multiple independent (not correlated) forecast models must be used to represent this characteristic.

- *Short-Term and Long-Term Forecasts:* Furthermore, the separate processes of one-day-ahead planning and intra-day re-scheduling require a forecast model that allows for very accurate very-short- and short-term forecasts (with low time granularity of 15 min units) and accurate long-term forecasts.

- *High Update Rates:* Forecasting of energy demand and supply has another fundamental property. There is a very high update rate because one can measure the actual total consumption and the amount of requested energy at arbitrary granularity (with regard to all aggregation dimensions), where all these measurements exhibit an append-only characteristic. The advantage for forecasting is that the updates offer continuous feedback. A challenge results from the high update rate, because it poses high efficiency challenges. As a result, this dynamic of many updates should be taken into account for defining efficient maintenance strategies of forecast models.

- *Hierarchical Data Warehouse:* Due to the regionally distributed actors of the energy demand and supply balancing process, the data management architecture exhibit the character of a hierarchical data warehouse that is physically distributed in order to allow for high scalability. This characteristic requires the distributed maintenance of forecast models and their synchronization in order to guarantee the convergence of forecasts.

6.2 Research Challenges

Both energy data characteristics and the characteristics of the specific process of balancing energy demand and supply pose major research challenges. In the following, we briefly describe these challenges and outline interesting directions for future work in more detail.

- *Parameterless Forecasting:* Due to the different application areas of energy demand and supply forecasting as well as the presence of arbitrary aggregation hierarchies, the forecast model should be parameterless in the sense that no parameters, thresholds and influencing factors are required to be specified by the user. As a result, a forecast model, maintenance and aggregation strategies are required that enable high accurate and efficiency without the need of human intervention by trying to compute optimal or near-optimal parameters with regard to the given data.

- *Update Feedback Exploitation:* The combination of the characteristic of aggregation-awareness and high update-rates lead to the challenge of exploiting this continuous feedback in order to enhance the accuracy of the current forecast model. This implies the evaluation of an appropriate error metric. In the presence of long time series it is important to define the scope of the error metric in order to determine drifts as soon as possible.

- *Incremental Maintenance (Efficient Parameter Re-Estimation):* The high update rate in the sense of measured observations implies that it is impossible to re-estimate the parameters of the existing forecast model for ever update. However, the huge number of updates might rapidly change the time series such that the estimated parameters are not optimal leading to inaccurate forecasts. As a result, there is the challenge of incremental maintenance of forecast models. This includes the three steps of model identification (choosing the order of the model), model estimation (determining the optimal parameters), and the incremental maintenance of forecast values. While the model identification might result in static decisions and the maintenance of forecast values is not expensive, the major research question is if and how we could incrementally maintain model parameters or if we can at least determine when to trigger re-estimation.

- *Distributed Forecasting:* The hierarchy of data warehouses in combination with the requirement of aggregation-awareness leads to the challenge of distributed forecasting. Essentially, one must ensure convergence of global forecast models with all (recursive) local forecast models. This challenge is strengthened in the presence of high update rates, incremental maintenance and all exogenous influences that depend strongly on the local properties.

According to our review of related work, there are plenty of results on forecasting energy demand and some approaches of forecasting supply as well. Unfortunately, there are no overall approaches for distributed forecasting with regard to efficient maintenance of forecast models on different aggregation levels. The major challenge, in this domain, is the efficient maintenance of forecast models over evolving time series. This challenge is strengthen in the context of a distributed aggregation hierarchy with forecast models at each hierarchy level.

7 Conclusions

In this survey, we reviewed forecast models for balancing energy demand and supply. The major goals in the energy domain are the active (real-time) customer involvement and the integration of more renewable energy sources that cannot be planned. Both goals inherently lead to the challenge of the balancing of energy demand and supply. In this context, accurate and efficient forecasting is a fundamental precondition for enabling robust balancing or scheduling of demand and supply.

We reviewed general time series forecasting techniques and classified existing forecast models for energy demand and supply; we also categorized them according to their seasonality and mathematical characteristics. In general, there is plenty of related work on energy demand forecasting (single-equation and multi-equation) but there are only few approaches for energy supply forecasting. The latter indeed is a recent problem triggered by the increasing supply from renewable energy sources, while the traditional energy sources can be planned more easily. Furthermore, it is important to note that there are no approaches for distributed forecasting over an organizational system hierarchy. In addition, we revealed typical energy data characteristics and evaluated general-purpose forecast models as well as forecast models that are tailor-made for energy demand and supply. Here, we found that especially multi-equation forecast models achieve very high accuracy for both very-short-term as well as long-term forecast horizons. Finally, we pointed out major characteristics and research challenges in this important domain of energy demand and supply forecasting.

In conclusion, there is a huge variety of forecast models for energy demand that achieve high accuracy for different forecast horizons. However, most of these models are designed for a static analysis of time series. The real-time requirements of the changing energy market created by the goals of active customer involvement as well as the integration of more renewable energy sources additionally require efficient forecast model maintenance over evolving time series in distributed system architectures. Thus, especially from a system architectural perspective, major research challenges exist in the area of forecasting the energy demand and supply.

Bibliography

[ACU10] Mert Akdere, Ugur Cetintemel, and Eli Upfal. Database-support for continuous prediction queries over streaming data. In *VLDB*, 2010.

[ASS08] Luiz Felipe Amaral, Reinaldo Castro Souza, and Maxwell Stevenson. A smooth transition periodic autoregressive (stpar) model for short-term load forecasting. *International Journal of Forecasting*, 24(4):603 – 615, 2008.

[BBS88] H.A.P. Blom and Y. Bar-Shalom. The interacting multiple model algorithm for systems with markovian switching coefficients. *IEEE Transaction on Automatic Control*, 33:780–783, 1988.

[BD91] Peter J. Brockwell and Richard A. Davis. *Time Series: Theory and Methods.* Springer Verlag Inc., 1991.

[BD02] Peter J. Brockwell and Richard A. Davis. *Introduction to Time Series and Forecasting.* Springer Verlag Inc., 2002.

[BJR08] George E. P. Box, Gwileym M. Jenkins, and Gregory C. Reinsel. *Time Series Analysis: Forecasting and Control.* John Wiley & Sons Inc., 2008.

[BKPM08] Marek Brabec, Ondrej Konár, Emil Pelikán, and Marek Malý. A nonlinear mixed effects model for the prediction of natural gas consumption by individual customers. *International Journal of Forecasting*, 24(4):659 – 678, 2008.

[BLNZ95] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM J. Scientific Computing*, 16:1190–1208, 1995.

[Bun00] Derek W. Bunn. Forecasting loads and prices in competitive power markets. In *Proceedings of the IEEE, VOL. 88, NO. 2*, 2000.

[Bur98] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[BW09] E. Borghers and P. Wessa. Statistics - econometrics - forecasting. Technical report, Office for Research Development and Education, 2009.

[CCC$^+$05] B.P. Challa, S. Challa, R. Chakravorty, S.K. Deshpande, and D. Sharma. A novel approach for electrical load forecasting using distributed sensor networks. In *Proceedings of the Third International Conference on Intelligent Sensing and Information (ICISIP)*, pages 189 –194, Dec. 2005.

[CCSG02] Antonio S. Cofino, Rafael Cano, Carmen Sordo, and Jose M. Gutierrez. Bayesian networks for probabilistic weather prediction. In *In Proceedings of the 15th European Conference on Artificial Intelligence*, 2002.

[CDD$^+$09] Jeffrey Cohen, Brian Dolan, Mark Dunlap, Joseph M. Hellerstein, and Caleb Welton. Mad skills: New analysis practices for big data. *PVLDB*, 2(2):1481–1492, 2009.

[CDH10] Raja Chiky, Laurent Decreusefond, and Georges Hébrail. Aggregation of asynchronous electric power consumption time series knowing the integral. In *EDBT*, 2010.

[CEC03] Protocol development for demand response calculation - findings and recommendations. Technical report, California Energy Commission, 2003.

[CEG08] José Ramón Cancelo, Antoni Espasa, and Rosmarie Grafe. Forecasting the electricity load from one day to one week ahead for the spanish system operator. *International Journal of Forecasting*, 24(4):588 – 602, 2008.

[CG09] Xuefeng Chen and Chaozhen Guo. The research on power marketing dss based on distributed collaborative forecasting model. In *Proceedings of the 2009 13th International Conference on Computer Supported Cooperative Work in Design*, 2009.

[CS03] Remy Cottet and Michael Smith. Bayesian modeling and forecasting of intraday electricity load. *Journal of the American Statistical Association*, 98:839–849, 2003.

[CSG04] Rafael Cano, Carmen Sordo, and Jose M. Gutierrez. Applications of bayesian networks in meteorology. *Advances in Bayesian Networks*, 1:309–327, 2004.

[CT86] K.S. Chan and H. Tong. On estimating thresholds in autoregressive models. *Journal of Time Series Analysis*, 7:178–190, 1986.

[DB07] Songyun Duan and Shivnath Babu. Processing forecasting queries. In *VLDB*, 2007.

[DCSW09] Umeshwar Dayal, Malú Castellanos, Alkis Simitsis, and Kevin Wilkinson. Data integration flows for business intelligence. In *EDBT*, pages 1–11, 2009.

[DKO$^+$08] V. Dordonnat, S.J. Koopman, M. Ooms, A. Dessertaine, and J. Collet. An hourly periodic state space model for modelling french national electricity load. *International Journal of Forecasting*, 24(4):566 – 587, 2008.

[DLL03] Serge Degerine and Sophie Lambert-Lacroix. Characterization of the partial autocorrelation function of nonstationary time series. *Journal of Multivariate Analysis*, 87:46–59, 2003.

[DSB$^+$10] Sudipto Das, Yannis Sismanis, Kevin S. Beyer, Rainer Gemulla, Peter J. Haas, and John McPherson. Ricardo: integrating r and hadoop. In *SIGMOD Conference*, pages 987–998, 2010.

[dSFV08] Alexandre P. Alves da Silva, Vitor H. Ferreira, and Roberto M.G. Velasquez. Input space to neural network based load forecasters. *International Journal of Forecasting*, 24(4):616 – 629, 2008.

[Esh09] Gidon Eshel. The yule walker equations for the ar coefficients. Technical report, Bard College at Simon's Rock, 2009.

[Fle08] Tristan Fletcher. Support vector machines explained. Technical report, University College London, 2008.

[GKO$^+$08] Phillip G. Gould, Anne B. Koehler, J. Keith Ord, Ralph D. Snyder, and Rob J. Hyndman Farshid Vahid-Araghi. Forecasting time series with multiple seasonal patterns. *European Journal of Operational Research*, 191:207–222, 2008.

[GL99] Paul Goodwin and Richard Lawton. On the asymmetry of the symmetric mape. *International Journal of Forecasting 15*, 15:405–408, 1999.

[Gun98] Steve R. Gunn. Support vector machines for classification and regression. Technical report, University of Southampton, 1998.

[GZ08] Tingjian Ge and Stan Zdonik. A skip-list approach for efficiently processing forecasting queries. In *VLDB*, 2008.

[Här00] Wolfgang Härdle. Tutorials for statistics of financial markets. Technical report, Humboldt University Berlin, 2000.

[Hyn06] Rob J. Hyndman. Another look at forecast-accuracy metrics for intermittent demand. *Foresight: The International Journal of Applied Forecasting*, 4:43–46, 2006.

[JR08] René Jursa and Kurt Rohrig. Short-term wind power forecasting using evolutionary algorithms for the automated specification of artificial intelligence models. *International Journal of Forecasting*, 24(4):694 – 709, 2008.

[KB08] Nektaria V. Karakatsani and Derek W. Bunn. Forecasting electricity prices: The impact of fundamentals and time-varying coefficients. In *International Journal of Forecasting 24*, 2008.

[Kec01] Vojislav Kecman. *Learning and Soft Computing: Support Vector Machines, Neural Networks and Fuzzy Logic Models*. The MIT Press, 2001.

[LGS06] Ming Li, Deepak Ganesan, and Prashant Shenoy. Presto: Feedbackdriven data management in sensor networks. In *In Proceedings of the NSDI 06: 3rd Symposium on Networked Systems Design & Implementation*, 2006.

[McC98] B.D. McCullough. Algorithm choice for (partial) autocorrelation functions. *Journal of Economic and Social Measurement*, 24:265–278, 1998.

[MEK06] Kourosh Mohammadi, H.R. Eslami, and Rene Kahawita. Parameter estimation of an arma model for river flow forecasting using goal programming. *Journal of Hydrology*, 331(1-2):293 – 299, 2006.

[Mic07] Microsoft. *Microsoft SQL Server 2008 - Analysis Services Overview*, 2007.

[Mic08] Microsoft. *Microsoft SQL Server 2008 - Predictive Analysis with SQL Server 2008*, 2008.

[MNG05] Amit Manjhi, Suman Nath, and Phillip B. Gibbons. Tributaries and deltas: Efficient and robust aggregation in sensor network streams. In *SIGMOD Conference*, pages 287–298, 2005.

[MSR$^+$97] Klaus-Robert Müller, Alex J. Smola, Gunnar Rätsch, Bernhard Schölkopf, Jens Kohlmorgen, and Vladimir Vapnik. Predicting time series with support vector machines. In *In Proceeding of the 7th International Conference on Artifical Neural Networks*, 1997.

[Nak94] Gholamreza Nakhaeizadeh. Learning prediction of time series. a theoretical and empirical comparison of cbr with some other approaches. In *Topics in Case-Based Reasoning*, pages 65–76, 1994.

[Nat10] NationalGrid. *nationalgrid UK - Metered half-hourly electricity demands*, 2010. `http://www.nationalgrid.com/uk/Electricity/Data/Demand+Data/`.

[NCR09] Manoel C. Amorim Neto, George D. C. Calvalcanti, and Tsang Ing Ren. Financial time series prediction using exogenous series and combined neural networks. In *Proceedings of International Joint Conference on Neural Networks, Atlanta, Georgia, USA*, 2009.

[NIS10] NIST/SEMATECH. e-handbook of statistical methods. Technical report, National Institute of Standards and Technology, 2010.

[O'C08] William O'Connell. Extreme streaming: business optimization driving algorithmic challenges. In *SIGMOD Conference*, pages 13–14, 2008.

[Ora08] Oracle. *Oracle - Driving Strategic Planning with Predictive Modeling*, 2008.

[PESI$^+$91] D.C. Park, M.A. El-Sharkawi, R.J. Marks II, L.E. Atlas, and M.J. Damborg. Electric load forecasting using an artifical neural network. *IEEE Transactions on Power Systems*, 6:442–450, 1991.

[PH05] Ping-Feng Pai and Wei-Chiang Hong. Support vector machines with simulated annealing algorithms in electricity load forecasting. *Energy Conversion and Management*, 46:2669–2688, 2005.

[Pru04] H. Pruys. *Die Maximum-Likelihood-Methode.* Institut für Physik Universität Zürich, 2004.

[R10] *The R Project for Statistical Computing*, 2010.

[REG$^+$97] Ramu Ramanathan, Robert Engle, Clive W.J. Granger, Farshid Vahid-Araghi, and Casey Brace. Short-run forecasts of electricity loads and peaks. *International Journal of Forecasting*, 13:161–174, 1997.

[RP87] G. Rebane and J. Pearl. The recovery of causal poly-trees from statistical data. In *Workshop on Uncertainty in AI*, pages 222–228, 1987.

[SBdM07] Reinaldo Castro Souza, Mônica Barros, and Cristina Vidigal C. de Miranda. Short term load forecasting using double seasonal exponential smoothing and interventions to account for holidays and temperature effects. Technical report, Pontifícia Universidade Católica do Rio de Janeiro, 2007.

[SM08] Lacir J. Soares and Marcelo C. Medeiros. Modeling and forecasting short-term electricity load: A comparison of methods with an application to brazilian data. *International Journal of Forecasting*, 24(4):630 – 644, 2008.

[Sán08] Ismael Sánchez. Adaptive combination of forecasts with application to wind energy. *International Journal of Forecasting*, 24(4):679 – 693, 2008.

[SS98] Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. Technical report, NeuroCOLT2 - ESPRIT Working Group, 1998.

[Sto07a] Herbert Stocker. *Angewandte Ökonometrie: Die Maximum-Likelihood Methode.* Institut für Wirtschaftstheorie, -politik und -geschichte Universität Insbruck, 2007.

[Sto07b] Herbert Stocker. *Angewandte Ökonometrie: Multiple Regression.* Institut für Wirtschaftstheorie, -politik und -geschichte Universität Insbruck, 2007.

[Tay09] James W. Taylor. Triple seasonal methods for short-term electricity demand forecasting. *European Journal of Operational Research*, 204:139–152, 2009.

[TdMM06] James W. Taylor, Lilian M. de Menezes, and Patrick E. McSharry. A comparison of univariate methods for forecasting electricity demand up to a day ahead. *International Journal of Forecasting*, 22:1–16, 2006.

[TM08] J. W. Taylor and P. E. McSharry. Short-term load forecasting methods: An evaluation based on european data. *IEEE Transactions on Power Systems*, 22:2213–2219, 2008.

[US 10] US Energy Information Administration, Independent Statistics and Analysis. *US EIA - International Energy Statistics*, 2010. `http://tonto.eia.doe.gov/cfapps/ipdbproject/IEDIndex3.cfm?tid=2&pid=2&aid=2`.

[Vap98] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[WK10] Richard Winter and Pekka Kostamaa. Large scale data warehousing: Trends and observations. In *ICDE*, page 1, 2010.

[WM08] Rafal Weron and Adam Misiorek. Forecasting spot electricity prices: A comparison of parametric and semiparametric time series models. *International Journal of Forecasting*, 24:744–763, 2008.

[WMHM01] G Winkler, C Meisenbach, M Hable, and P Meier. Intelligent energy management of electrical power systems with distributed feeding on the basis of forecasts of demand and generation. In *CIRED*, 2001.

[Zac03] Dr. John Zack. Overview of wind energy generation forecasting. Technical report, TrueWind Solutions, LLC, 2003.

[ZSY04] Changshui Zhang, Shiliang Sun, and Guoqiang Yu. A bayesian network approach to time series forecasting of short-term traffic flows. In *In Proceedings of the 2004 IEEE Intelligent Transportation Systems Conference*, 2004.