FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

ASSIGNMENT OF MASTER’S THESIS

Title:	Electricity price forecasting based on weather conditions
Student:	Bc. Jan Dejdar
Supervisor:	Ing. Kamil Dedecius, Ph.D.
Study Programme:	Informatics
Study Branch:	Knowledge Engineering
Department:	Department of Applied Mathematics
Validity:	Until the end of summer semester 2019/20

Instructions

Diploma thesis deals with electricity prices modelling and forecasting based on weather conditions. In general, electricity prices, supply and demand have a very volatile character due to the impossibility to store electricity, the need of a constant balance between its production and consumption, and its dependence on weather. For instance, the supply of photovoltaic and wind power plants is highly reliant on weather conditions regarding exposure to sunlight and wind, respectively. On the other hand, the demand is connected with the temperature, daytime, season etc. All these phenomena are continually monitored, and their reliable modelling is of great interest.

Diploma thesis instructions:

1) Analyze available electricity prices and weather data. Use any appropriate measures, preferably statistical ones.
2) Propose potentially convenient models for these data. Perform their validation and evaluation. Focus on their predictive performance.
3) Discuss the obtained results.

References

Will be provided by the supervisor.

Ing. Karel Klouda, Ph.D.
Head of Department

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
Dean

Prague December 14, 2018

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Master's thesis

Electricity price forecasting based on weather conditions

Bc. Jan Dejdar

Department of Applied Mathematics
Supervisor: Ing. Kamil Dedecius, Ph.D.

May 8, 2019

Acknowledgements

I would like to thank Ing. Kamil Dedecius, Ph.D. for all the time he spent helping me with this thesis, for all of his advice and his patience. I would also like to thank Bohuslav Zeman for his consultations and EEX Group for providing the electricity price data for this thesis. Last but not least, I would like to thank my girlfriend Veronika and my family for their support during my studies.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as school work under the provisions of Article 60(1) of the Act.

In Prague on May 8, 2019 ....

Citation of this thesis

Dejdar, Jan. *Electricity price forecasting based on weather conditions.* Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2019.

Abstrakt

Tato diplomová práce se zabývá předpovídáním ceny elektřiny v závislosti na počasí. Výzkum tohoto tématu je předmětem velkého zájmu, protože na přesných předpovědích cen elektřiny závisí provozovatelé elektráren i velké podniky, které elektřinu spotřebovávají. Vzhledem k nemožnosti efektivního ukládání většího množství elektřiny a nutnosti vyvažovat v síti mezi produkcí a spotřebou cena elektřiny na krátkodobých trzích velmi výrazně kolísá, což činí její přesné předpovídání náročným.

Cenu elektřiny ovlivňuje mnoho faktorů. Zejména ve střední a západní Evropě je jedním z nich počasí, kvůli stále se zvyšujícímu množství solárních a větrných elektráren. Cílem této práce je analyzovat dostupná data o počasí a cenách elektřiny na burze a vytvořit pro tato data prediktivní model.

Klíčová slova předpověď cen elektřiny, časové řady, počasí

Abstract

This thesis deals with electricity price forecasting based on weather conditions. The research of this topic is of great interest because accurate forecasting is essential for both the electricity producers and consumers. Due to the inability of storing a large amount of electricity efficiently and the necessity to balance between production and consumption, the short-term electricity price tends to be very volatile, which makes it difficult to predict it accurately.

There are many factors that affect the electricity price. Especially in Central and Western Europe, very important factor is the weather, due to increasing number of solar and wind power plants. This thesis aims to analyze available data about weather forecasts and the electricity prices and create predictive models for this data.

Keywords electricity price forecasting, time series, weather

Contents

List of Figures

List of Tables

Introduction

Ensuring a stable electricity supply is a very complicated task. It is not yet possible to store a large amount of electricity efficiently. On the other hand, it is critical to assure that the electricity is available in the network all the time. Both the demand and supply vary greatly over time. The demand is generally lower in the night and higher during the daytime, but it also changes seasonally on a weekly, monthly and yearly basis. Part of the supply is created by renewable power sources, such as wind or photovoltaic power plants, where the supply is determined by the current weather - especially the wind conditions and the cloud cover - and therefore can change rapidly.

Some form of a regulator that ensures that the supply and demand in the transmission grid are balanced exists in most regions. If more electricity is supplied to the grid than taken out, the utility frequency in the system can increase (and vice versa), which can cause serious problems. The regulator is responsible for keeping reserves to compensate for possible disruptions in the supply. Until a few decades ago, the state-owned companies had a monopoly in the electricity market in every country in the world. The first country to attempt privatization of the electric power systems and subsequent creation of a deregulated power market was South American Chile in the early 1980s [1]. This attempt is generally considered to be fairly successful. After Chile, several other South American countries tried to privatize this industry to some extent, due to the necessity of investments in the infrastructure. However, the results were limited.

In 1990, UK's Prime Minister Margaret Thatcher privatized electricity supply industry in the United Kingdom [2]. During the first few years, the electricity prices rose, but since 2000, the consumers started to gain from the transformation. Consequently, other Commonwealth countries adopted the British model into their economies.

In the United States of America, about a dozen states implemented deregulation of the power market in the late 1990s. In 2000–2001, California experienced vast electricity supply shortages and consequent rapid increase in electricity prices. The price for a megawatt hour rose from $30 in April of 2000 as high as $450 in November of the same year. This rise was caused mainly by a combination of increased demand, reduction in imports, and delays in new power plants approvals [3].

Many developed countries chose to deregulate the electricity market to some extent, allowing private companies to buy and sell the electricity on the market. Nowadays, it can be traded intra-day (balance market), day-ahead (spot market), or for a longer time ahead (futures market). This thesis aims to create a predictive model for the spot market prices. The short-term market is convenient for this purpose because it is very liquid and therefore the time series of the prices is available. Also, reliable weather forecasts are not available over a longer period of time, and therefore the price in the futures market is less influenced by the weather than the spot market prices.

Chapter 1

Theoretical background

In this chapter, the theoretical background of the concepts used in the thesis is presented. These include both the models used for the forecasting and statistical tools for their analyses and evaluation.

1.1 Likelihood function

Let $X_1, \dots, X_n$ be a random sample, $x_1, \dots, x_n$ the values of the random sample from a population characterized by the parameters $\theta_1, \dots, \theta_r$. According to [4], the likelihood function of the sample is defined as the joint probability mass function evaluated at $(x_1, \dots, x_n)$ if $(X_1, \dots, X_n)$ are discrete,

$$L\left(\theta_1, \dots, \theta_r\right) = p\left(x_1, \dots, x_n; \theta_1, \dots, \theta_r\right), \tag{1.1}$$

or the joint probability density function evaluated at $(x_1, \dots, x_n)$ if $(X_1, \dots, X_n)$ are continuous,

$$L\left(\theta_1, \dots, \theta_r\right) = f\left(x_1, \dots, x_n; \theta_1, \dots, \theta_r\right). \tag{1.2}$$

The value of the likelihood function is a positive number, often very small. This fact makes it convenient to work with a logarithmic transformation of the likelihood function, known as the log-likelihood function.

1.2 Akaike information criterion

Akaike information criterion (AIC) is a statistical tool that can be used to estimate the relative quality of statistical models for a given dataset. It was formulated by Japanese statistician Hirotugu Akaike in 1973 [5].

AIC can be computed as

$$AIC = -2\log\left(L\left(\hat{\theta}|y\right)\right) + 2K, \tag{1.3}$$

where $\log\left(L\left(\hat{\theta}|y\right)\right)$ is the numerical value of the log-likelihood at its maximum point and K denotes the number of estimable parameters in the model, according to [6]. The models with lower value of AIC are considered to be better than those with higher value.

1.3 Distributions

In the following sections, the distributions that are used later in the thesis are defined.

1.3.1 Bernoulli distribution

Bernoulli distribution is a discrete distribution of a random variable X that can take two possible values with probabilities p and $1-p$. Let X be a random variable that satisfies

$$P\left(X=1\right)=p \tag{1.4}$$

and

$$P\left(X=0\right)=1-p. \tag{1.5}$$

Random variable X then has the Bernoulli distribution with parameter p, $X \sim Bernoulli(p)$ [4].

1.3.2 Categorical distribution

Categorical distribution is a generalization of the Bernoulli distribution to K possible outcomes. Let x be K-dimensional vector in which one of its elements x_k is equal to 1 and all remaining elements are equal to 0, and $\mu = (\mu_1, \ldots, \mu_K)$ is a vector containing the probabilities of $x_1, \ldots, x_K$ being equal to 1. According to [7], categorical distribution is then given by

$$p\left(x|\mu\right)=\prod_{k=1}^{K}\mu_k^{x_k}. \tag{1.6}$$

The elements of the vector $\mu = (\mu_1, \ldots, \mu_K)$ represent probabilities and therefore satisfy conditions

$$\sum_{k=1}^{K}\mu_k = 1 \tag{1.7}$$

and

$$\mu_k \geq 0, k \in \left\{1, \ldots, K\right\}. \tag{1.8}$$

1.3.3 Dirichlet distribution

Let $\mu = (\mu_1, \ldots, \mu_K)$ be vector such that $\sum_{k=1}^{K} \mu_k = 1$ and $\mu_k \in [0,1]$ for each $k \in \{1, \ldots, K\}$. According to [7], the Dirichlet distribution is given by

$$Dir\left(\mu|\alpha\right) = \frac{\Gamma\left(\sum_{k=1}^{K} \alpha_k\right)}{\prod_{k=1}^{K} \Gamma\left(\alpha_k\right)} \prod_{k=1}^{K} \mu_k^{\alpha_k - 1}, \tag{1.9}$$

where Γ is the gamma function and $\alpha = (\alpha_1, \ldots, \alpha_K)$ are the parameters of the distribution.

1.4 Weak stationarity

Weak stationarity is an important property of time series that is necessary for some models. According to [8], the time series is weakly stationary if its expected value and variance are time-invariant, and the variance is finite.

1.5 Correlation

In the broadest sense, correlation is a statistical association between two random variables. It is usually used to establish how much two random variables are related. In the following sections, some of the applications of this concept are discussed.

1.5.1 Pearson correlation coefficient

Pearson correlation coefficient (PCC) is a statistical measure of mutual linear dependence between random variables X and Y. Assuming positive variances of random variables X, Y, Pearson correlation coefficient $\rho_{X,Y}$ can be mathematically expressed as

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_x \sigma_y} = \frac{\operatorname{cov}(X,Y)}{\sqrt{varX}\sqrt{varY}}. \tag{1.10}$$

PCC can be applied to a data sample by replacing variances and covariances with their estimates in Equation (1.10). For n samples $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, sample Pearson correlation coefficient estimate can be expressed as

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 (y_i - \overline{y})^2}}, \tag{1.11}$$

where $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ is the sample mean of x ($\overline{y}$ can be calculated analogously).

The value of the PCC is a real number from closed interval $[-1, 1]$ where value -1 means total negative linear relationship, 0 no linear relationship at all and 1 total positive linear relationship. An example of the correlation coefficients for various X and Y pairs is visualized in Figure 1.1.

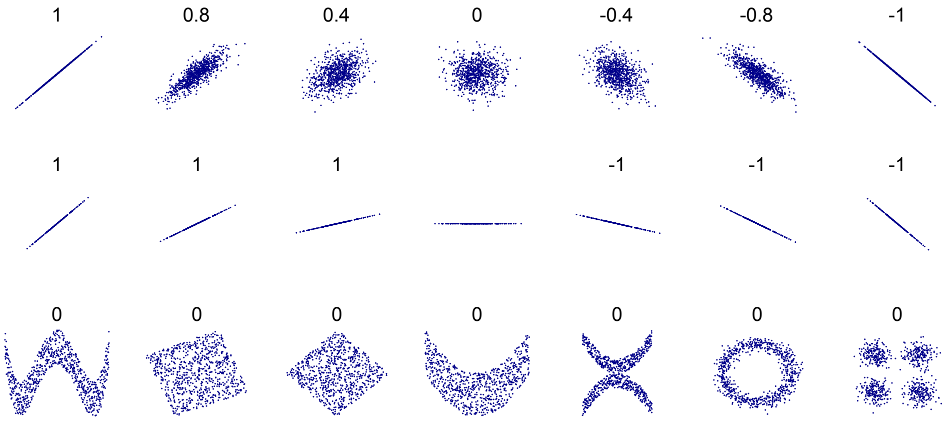


Figure 1.1: Examples of correlation coefficients [9]

1.5.2 Partial correlation coefficient

Let X and Y be random variables with some linear relationship between them, and Z n-dimensional vector that affects both X and Y. To measure the correlation between the two random variables, it is necessary to remove the influence of the vector Z. The partial correlation coefficient $\rho_{XY\cdot Z}$ between X and Y given Z is defined as

$$\rho_{XY\cdot Z} = \frac{\rho_{XY} - \operatorname{cor}(X,Z)(\operatorname{cor} Z)^{-1}\operatorname{cor}(Z,Y)}{\sqrt{[1-\operatorname{cor}(X,Z)(\operatorname{cor} Z)^{-1}\operatorname{cor}(Z,X)]\,[1-\operatorname{cor}(Y,Z)(\operatorname{cor} Z)^{-1}\operatorname{cor}(Z,Y)]}} \tag{1.12}$$

If Z is scalar, the above formula can be simplified to

$$\rho_{XY\cdot Z} = \frac{\rho_{XY} - \rho_{ZX}\rho_{ZY}}{\sqrt{(1-\rho_{ZX}^2)\,(1-\rho_{ZY}^2)}}. \tag{1.13}$$

The domain of the partial correlation coefficient is (just as for the Pearson correlation coefficient) a closed interval $[-1,1]$ [10].

1.5.3 Autocorrelation function

The autocorrelation function (ACF) for a time series is a Pearson correlation coefficient of the series in different points in time. For weakly stationary time series X_t and times s and t, ACF $R(s,t)$ is defined by

$$R(s,t) = \frac{\mathbb{E}[(X_t-\mu)(X_s-\mu)]}{\sigma^2}. \tag{1.14}$$

Assuming that the value of time series X_t is directly influenced by the previous value X_{t-1}, the absolute values of ACF will be gradually decreasing because the influence of the previous value decreases. ACF allows determining some properties of time series [11].

1.5.4 Partial autocorrelation function

Let X_t be weakly stationary time series. The partial autocorrelation function (PACF) $\alpha(k)$ of lag k is the autocorrelation between X_t and X_{t+k}, with removed linear relationship of values $X_{t+1}, \ldots, X_{t+k-1}$. It can be mathematically described as

$$\alpha(k) = r_{X_{t+k}X_t \cdot \{X_{t+1}, \ldots, X_{t+k-1}\}}. \tag{1.15}$$

PACF allows determining some properties of time series [11].

1.6 Root mean squared error

The root mean squared error (RMSE) is a popular and frequently used measure of differences between predictions and the real values. It is defined as a square root of the average of squared differences between observations and the model prediction. The measure is mathematically expressed as

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}, \tag{1.16}$$

where n is the number of predictions, y_t is the tth measurement and $\hat{y}_t$ is the tth prediction.

RMSE is always non-negative, with value 0 indicating a perfect model that always predicts the same value as an observation. It is used to compare the performance of different models on a particular dataset. It cannot be used to compare forecasting accuracy between datasets because it is dependent on the scale of the data [11].

1.7 White noise

The white noise is a random process X_t that satisfies the following conditions,

$$\mathbb{E}[X_t] = 0, \tag{1.17}$$

$$\mathrm{var}(X_t) = \sigma^2 < \infty, \tag{1.18}$$

$$\mathrm{cov}(X_t, X_{t+\tau}) = 0. \tag{1.19}$$

1.8 Lag operator

The lag operator (or backshift operator) L is used for convenient notation in many time series related concepts [11]. It is an operator that for a given element of a time series returns the previous element,

$$LX_t = X_{t-1}, \qquad t > 1. \tag{1.20}$$

The inverse of the lag operator L^{-1}, given X_t, returns the next value in the time series,

$$L^{-1}X_t = X_{t+1}. \tag{1.21}$$

The lag operator can be raised to an integer power, resulting in its multiple application,

$$L^k X_t = \underbrace{L \cdot L \cdots L}_{k\times} X_t = X_{t-k}. \tag{1.22}$$

1.9 Difference operator

The difference operator ∇ applied to an element of a time series returns the difference of this element with the previous one, which can be conveniently written using the lag operator L [11],

$$\nabla X_t = X_t - X_{t-1} = (1 - L)X_t. \tag{1.23}$$

Raising the operator to an integer power results in multiple application of the operator, which can also be simplified using the lag operator [11],

$$\nabla^k X_t = (1 - L)^k X_t. \tag{1.24}$$

1.10 ARIMA models

The ARIMA models were popularized by statisticians George Box and Gwilym Jenkins [10], and they have become one of the most popular models in time series modeling. ARIMA stands for AutoRegressive Integrated Moving Average, and it consists of two parts - the autoregressive (AR) part and the moving average (MA) part. The description of the models in the following sections references [11].

1.10.1 AR models

The autoregressive (AR) models describe random process using its past realizations. Autoregressive model AR(p) with order p is defined as

$$X_t = \beta_0 + \sum_{i=1}^{p} \beta_i X_{t-i} + \varepsilon_t, \tag{1.25}$$

where $\beta = (\beta_0, \ldots, \beta_p)$ is a vector of regression coefficients and ε_t is white noise.

1.10.2 MA models

Assuming that ε_t is white noise, the moving average model MA(q) with order q is defined as

$$X_t = \mu + \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}, \tag{1.26}$$

where μ is the mean of the time series and $\theta = (\theta_1, \ldots, \theta_q)$ are the parameters of the model.

1.10.3 ARMA models

The autoregressive moving average (ARMA) models combine both AR and MA approaches. ARMA(p, q) model with AR order p and MA order q can be expressed as

$$X_t = c + \varepsilon_t + \sum_{i=1}^{p} \phi_i X_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}, \tag{1.27}$$

where c is a constant and $\phi = (\phi_1, \ldots, \phi_p)$ are the parameters of the AR part.

1.10.4 ARIMA models

The autoregressive integrated moving average models (ARIMA) are a generalization of ARMA models. For non-stationary time series, it is possible to replace the original data values with differences between neighboring values to eliminate the non-stationarity. This process is called differencing and can be applied one or multiple times. The number of times the differencing is performed is called the *order* of differencing.

ARIMA models can be defined using lag operator L as

$$\left(1 - \sum_{i=1}^{p} \phi_i L^i\right) X_t = \left(1 + \sum_{i=1}^{q} \theta_i L^i\right) \varepsilon_t. \tag{1.28}$$

1.10.5 SARIMA models

The seasonal ARIMA (SARIMA) is an extension of the previously defined ARIMA model, which takes seasonality into account. Time series often have patterns that periodically repeat in one or more cycles. For example, the outside temperature has daily cycles and yearly cycles. This property is called *seasonality.* When modeling time series, seasonality causes non-stationarity.

The differencing was previously used to remove non-stationarity in a time series. The same approach can be used in this case, differencing the values across the period of the seasonality,

$$\nabla_s X_t = (1 - L^s)X_t = X_t - X_{t-s}. \tag{1.29}$$

Model SARIMA$(p, d, q)(P, D, Q)_s$ with seasonality s is defined as

$$\Phi_P(L^s)\phi_p(L)\nabla_s^D \nabla^d X_t = \theta_q(L)\Theta_Q(L^s)\varepsilon_t, \tag{1.30}$$

where

- $\Phi_P(L^s)$ is seasonal AR operator with order P,
- ϕ_p is AR operator with order p,
- ∇_s^D is seasonal difference operator with order D,
- ∇^d is difference operator with order d,
- $\Theta_Q(L^s)$ is seasonal MA operator with order Q,
- $\theta_q(L)$ is MA operator with order q.

1.10.6 SARIMAX models

The SARIMAX is an extension of the previously defined SARIMA models that allows to include exogenous regressors into the model. It is particularly useful when the target variable is to some extent determined by one or more explanatory variables. For the purpose of this thesis, the weather conditions can be considered as the explanatory variables of the electricity price.

According to [12], the SARIMAX model can be expressed as

$$X_t = u_t + \sum_{i=1}^{N} \beta_t^{(i)} x_t^{(i)}, \tag{1.31}$$

where N is the number of exogenous regressors, $x_t^{(i)}$ is ith exogenous regressor, $\beta_t^{(i)}$ is coefficient of ith exogenous regressor, and u_t represents SARIMA model,

$$\Phi_P(L^s)\phi_p(L)\nabla_s^D \nabla^d u_t = \theta_q(L)\Theta_Q(L^s)\varepsilon_t. \tag{1.32}$$

1.11 Neural Networks

The artificial neural networks (ANNs) is a class of machine learning models that are inspired by biological neural networks. ANNs have become very popular recently, due to their ability to successfully model a large variety of problems. According to [13], the main benefits of neural networks are:

1. Nonlinearity

   The structure of ANN can be non-linear, which is very important for modeling difficult problems which are often inherently non-linear.

2. Input-Output Mapping

   The ANNs belong to a group of machine learning techniques called supervised learning (or learning with a teacher). These methods involve creating a function that maps the input to output, given example input data and desired output. The process of creating the function is called *learning* or *training*, and the example data are called the *training set*.

   In the learning process, the model is being given randomly chosen data from the training set, and it adjusts its inner parameters (e.g., synaptic weights in the case of the ANNs) to minimize the difference between its prediction and the desired output.

3. Adaptivity

   The artificial neural networks can easily adapt to the changes in the environment they are operating in. They can be retrained to take the changes that might occur into account, or they can even be designed to adjust the synaptic weights in real time.

4. Fault tolerance

   Due to the fact that the knowledge of ANN is distributed across the whole network, ANNs are robust in the sense that when a part of the network gets damaged, the network can still perform reasonably, rather than fail completely.

1.11.1 Neuron

A neuron is an elementary unit of the artificial neural networks. According to [13], the basic model of an artificial neuron consists of three parts:

1. Synapses

   The synapses are the connecting links that bring input signal into the neuron. Each synapse is assigned a weight which multiplies the signal brought to the neuron by the synapse.

2. Summing function

   The summing function sums the input signals weighted by the weights of the synapses that brought them into the neuron and a constant term called *bias*.

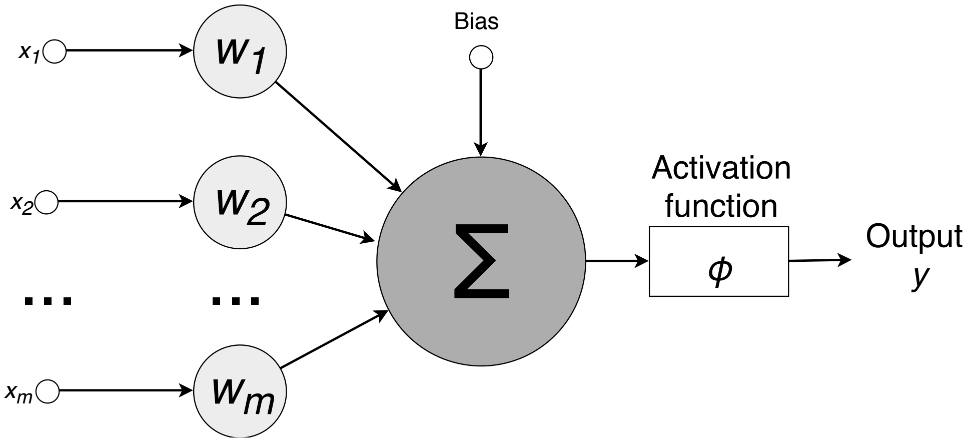


Figure 1.2: Basic model of artificial neuron

3. Activation function

   The activation function transform the output of the summing function into a defined interval - usually $[-1, 1]$ or $[0, 1]$. The function can be linear or non-linear.

Mathematically, neuron k can be described by equation

$$y_k = \varphi\left(\sum_{i=1}^{m} w_{ki} x_{ki} + b_k\right), \tag{1.33}$$

where φ is the activation function, $w_{k1}, \ldots, w_{km}$ are the synaptic weights of the neuron k, $x_{k1}, \ldots, x_{km}$ are the input signals, b_k is the bias term and y_k is the output of the neuron k. The basic model of an artificial neuron is shown in Figure 1.2.

According to [13], there are two main categories of activation functions:

1. Threshold functions

   Using this type of function, the output of the neuron is either 0 or 1. The threshold function can be for example the *Heaviside function*, which is given by

$$\varphi(v) = \begin{cases} 1 & \text{if } v \geq 0, \\ 0 & \text{if } v < 0. \end{cases} \tag{1.34}$$

2. Sigmoid functions

   The most common type of the activation functions are the sigmoid functions. They are strictly increasing and S-shaped. An example of the sigmoid function is the logistic function, which is defined as

$$\varphi(v) = \frac{1}{1 + e^{-av}}, \tag{1.35}$$

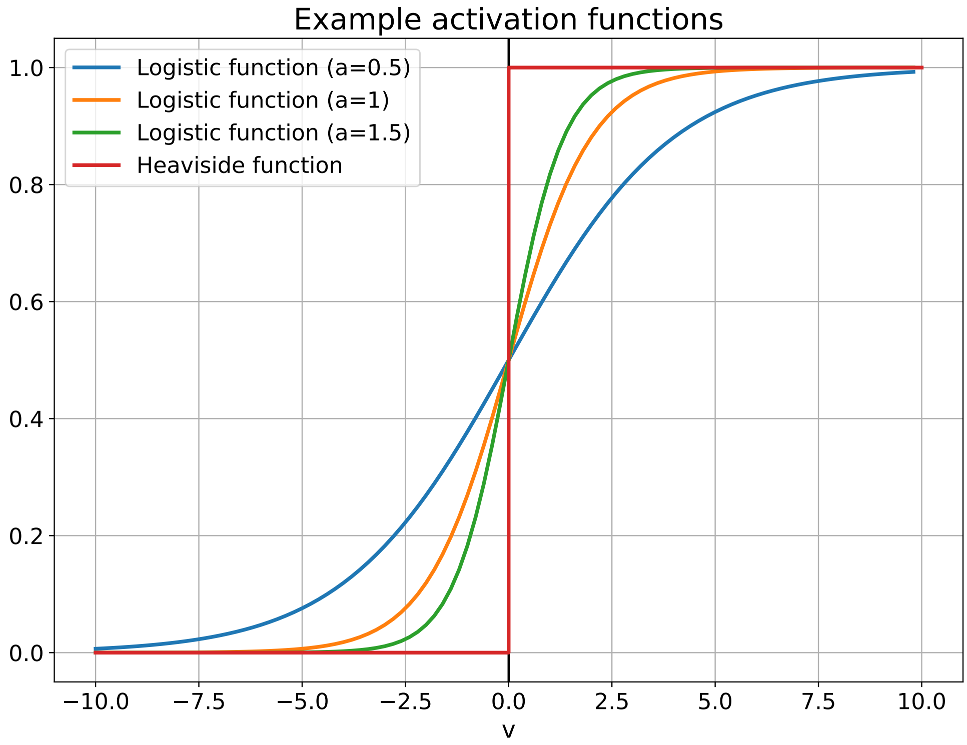


Figure 1.3: Various activation functions

where a is a parameter that affects the slope of the function. The effect of changing this parameter can be seen in Figure 1.3.

1.11.2 Network architectures

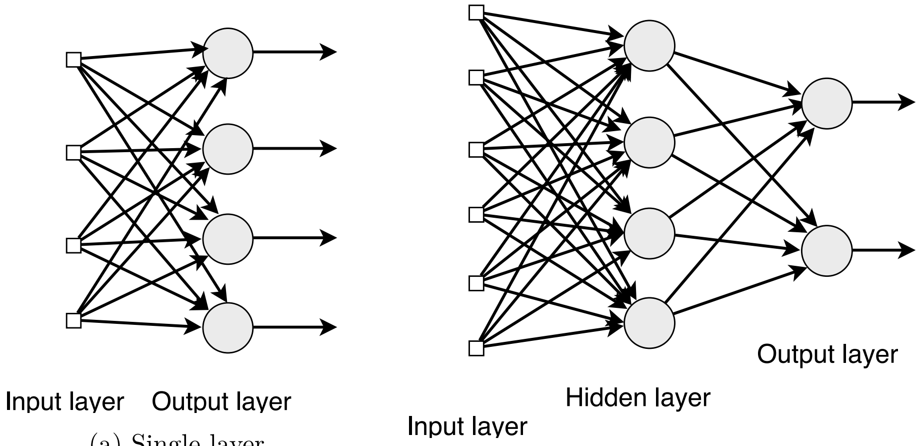
Multiple architectures of the artificial neural networks have been proposed. According to [13], three main types of ANN architecture can be distinguished:

1. Single-layer feedforward networks

   The most basic type of feedforward ANN is the single-layer network. It consists of one input layer of source nodes and one output layer of neurons. An example of the single layer ANN architecture with 4 output neurons is visualized in Figure 1.4a.

2. Multilayer feedforward networks

   The multilayer feedforward ANN contains, in addition to the single-layer ANN, one or more hidden layers of neurons. Using the hidden layers, the network is able to model more difficult problems, as the output of the neurons in the first layer is brought as the input to the neurons in



(a) Single-layer

(b) Multilayer

Figure 1.4: Feedforward networks

the second layer, and so on. An example of multilayer ANN with one hidden layer is visualized in Figure 1.4b.

3. Recurrent networks

   The recurrent neural networks (RNNs) contain one or more loops that bring the output of one or more neurons back to the input of some of the neurons. The RNNs can also be designed to have one or more hidden layers.

Chapter 2

Electricity price forecasting approaches

In this chapter, some of the existing techniques and approaches for electricity price forecasts are described.

Electricity price forecasting is a topic of great interest for many commercial researchers, as accurate forecasts are vital to both energy producers and consumers. With accurate forecasts, the energy producers, as well as large industrial costumers can adjust their market strategies accordingly to minimize the risks and maximize profits. Over the years, there have been many approaches to deal with this problem. Some of them are discussed in this chapter.

There are multiple general methods to tackle electricity price forecasting problem. According to [14, 15], the modeling techniques can be divided into five general categories, namely (i) multi-agent models, (ii) structural models, (iii) reduced-form models, (iv) statistical models, and (v) machine learning models.

2.1 Multi-agent models

These models are based on using multiple agents (electricity producers and consumers) that interact with each other in a simulation of the real market. The price is forecasted by matching the demand and supply in the simulated market. The inherent weakness of this approach is that it requires precise calibration parameters. Otherwise, it can diverge far from the real situation. The determination of these parameters is highly complicated.

An example model of this category can be the Cournot competition or the Supply function equilibrium (SFE) [15]. These two techniques were compared on the German electricity market in paper [16]. The researchers concluded that the SFE method is more robust than the Cournot competition, due to the fact that it relies less on the calibration parameters. However, they noted that the results for other than the German electricity market might be different.

2.2 Structural models

The structural models try to capture the relationships between the target variable (electricity price in this case) and the specific variables that influence it. These can include weather conditions, loads or specific system parameters.

This approach is very often applied in hybrid models that employ these exogenous variables to improve forecasting, in addition to exploiting other approaches. Purely structural models are usually more suitable for medium-term forecasts because the input data they need for the predictions are usually not available with hourly resolution [15].

2.3 Reduced-form models

These models aim to describe statistical properties of the target variable or correlations with other commodity prices, rather than to provide accurate hour-to-hour prices. [15]

2.4 Statistical models

The statistical models often use previous prices, or various exogenous regressors, e.g., weather for forecasting. Models belonging to this category can include, among others:

1. Similar-day methods

   This technique assumes that somehow similar days will also have similar electricity price. The similarity can be characterized by multiple factors - most importantly hour, day of the week, or other exogenous factors. The forecast can be computed as the price of the most similar day or as a combination of the prices of multiple similar days.

2. Regression models

   The regression models are popular for many problems. They are used to find relationships between the target variable and its exogenous regressors, which can, as in previously presented approaches, include, for example, hour, day of the week, or weather. This technique is widely used for EPF, but it is often combined with other approaches.

3. Autoregressive models

> This category includes models that use the previous values of the target variable to forecast new ones. They are very popular in time series forecasting. Numerous modifications and extensions of a simple autoregressive model exist, such as ARMA, ARIMA, SARIMA, and SARIMAX, which were defined in the previous chapter.

Probably the most popular statistical models for short-term electricity price forecasting belong to the ARIMA family. In paper [17], ARIMA models were tested on two different electricity markets - Spanish and Californian - to predict the prices for the next 24 hours. The achieved accuracy is different for each of the markets, with an average mean error around 10% for the Spanish market and around 5% for the Californian market. According to the authors, these results are good, compared with the performance of ANN, which they supposed to be a more suitable model for the data. The varying predictive performance for the two markets shows that it is not possible to compare the accuracy of the models across different markets, as they often behave differently. This is due to numerous reasons, the most important of which is probably different legislation in the US and Spain.

In a related study [18] carried out for the Californian market, the researchers discovered that modeling each hour separately improved the accuracy of the forecasts. This result was confirmed by paper [19], that performed the study on Leipzig Power Exchange using variously modified AR and ARMA models.

A paper examining the predictive power of weather for day-ahead electricity prices on the Scandinavian market was released in 2009 [20]. The researches concluded that the weather conditions helped to partially anticipate spikes in the prices.

2.5 Machine learning models

Machine learning models are a vast group of methods that were proposed to solve the limitations of the “classical” - e.g., probabilistic and statistical - models. They have become very popular recently. Their main advantage is that they can usually handle nonlinearity well. Techniques such as artificial neural networks (ANNs), decision trees or support vector machines (SVMs) belong to this category.

Artificial neural networks are often used for electricity price forecasting, and many papers dealing with exploiting them for EPF can be found. For example, in 2011 paper [21], an artificial neural network with two hidden layers was used to predict hourly electricity prices. The results revealed good performance during days with a normal trend but showed gradual degradation for days with significant price spikes.

Chapter 3

Analysis

In this chapter, an analysis of the electricity price forecasting problem is carried out. First, some of the electricity market mechanisms that are important for the forecasting are presented in Section 3.1. In Section 3.2, the particular electricity market for the modeling is selected. In Section 3.3, the available data sources, both for weather conditions and the electricity prices, are discussed. The selection of the approaches that are used for the modeling is described in the last Section 3.4.

3.1 Electricity market

Currently, the electricity market is divided into two main sections, namely the spot market and the futures market. In the spot market, the electricity is traded day-ahead, while in the futures market, the deals are closed for a longer period of time ahead - generally from a few days to a year.

In Figure 3.1, general schema of electricity spot market is visualized. During the day, the bids and offers for the electricity delivery for the next day are placed, until the market closes. For the European Energy Exchange (EEX) market, it is usually around noon CE(S)T. The market clearing price (MCP) is then established by the market operator as the intersection of the supply and demand curves.

On the spot market, the electricity price fluctuates significantly from hour to hour, usually having two peaks - one in the morning and the other in the evening. The minimal price mostly occurs in the night and the early afternoon. Typical price development over two days can be seen in Figure 3.2. However, the development can be very different during some anomalous conditions.

One of the very interesting things about the electricity price is the possibility of its negative value (in some markets, including the EEX). This usually happens because of two possible reasons - a very high supply or a very low demand (or a combination of the two). The high supply is tightly connected to the weather. It often happens when there is high electricity production from

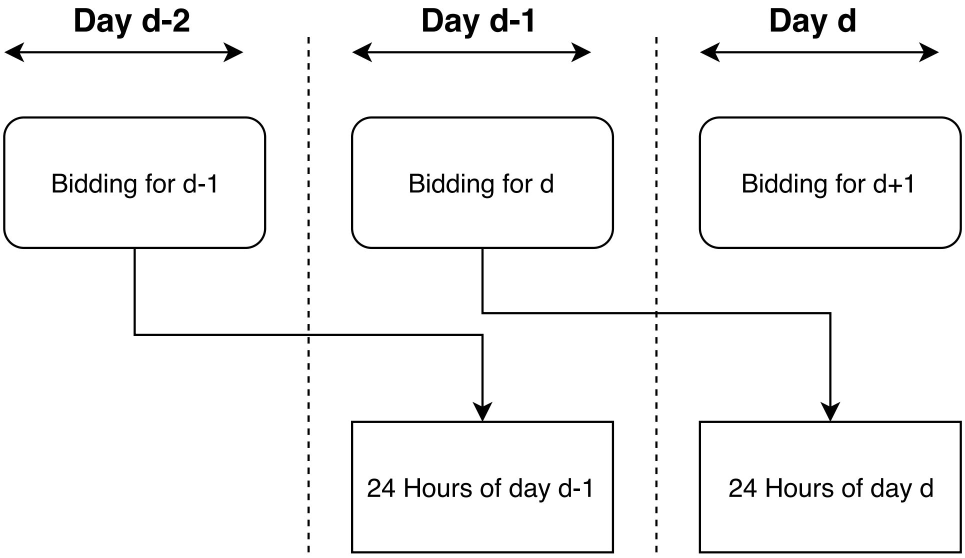


Figure 3.1: Electricity spot market schema

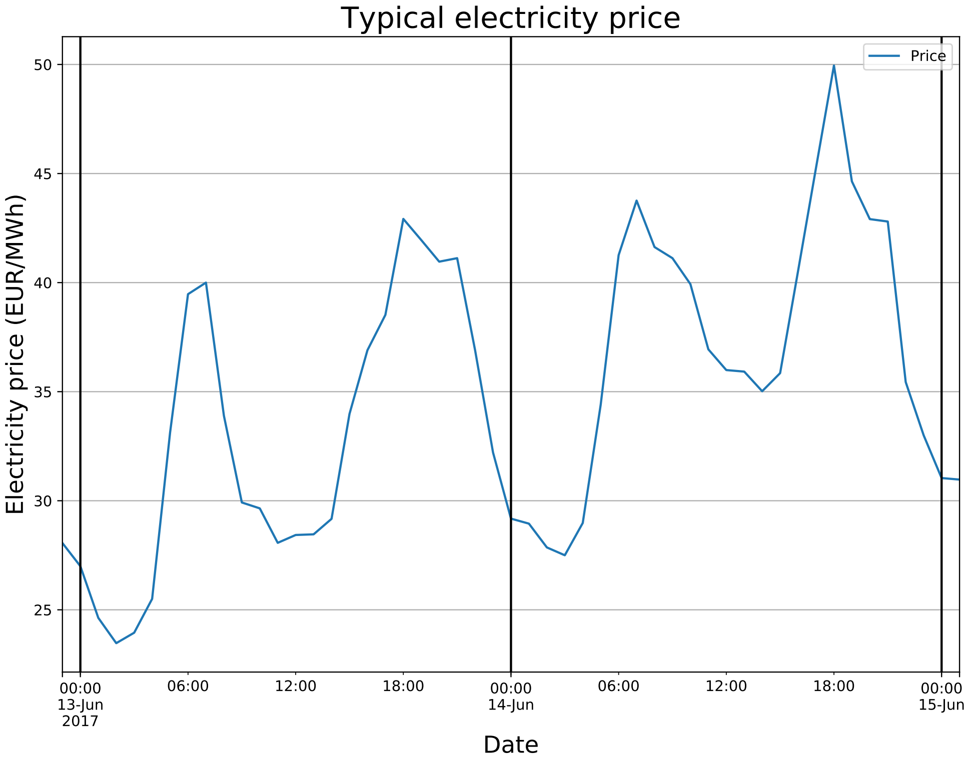


Figure 3.2: Typical electricity price development

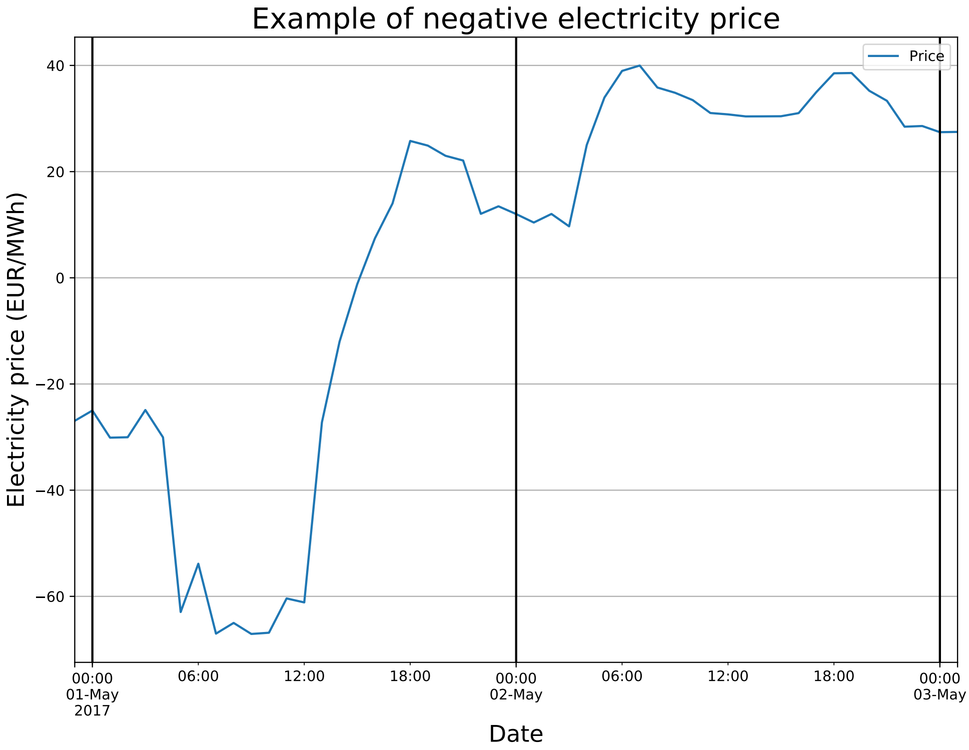


Figure 3.3: Example of negative electricity prices

renewable power plants, mainly the wind and solar power plants. On the other hand, low demand usually occurs during the night. The electricity producers often accept the occasional negative price, as it is cheaper than shutting the power plant down for a few hours. An example of negative electricity prices is shown in Figure 3.3.

3.2 Selection of the market for modeling

The electricity markets are usually national. The markets in different countries may vary, e.g., due to legislation and therefore it is necessary to choose a single national market for the forecasting. Because this thesis deals with forecasting based on weather conditions, the market should be in a country that generates a significant part of its electricity from solar and wind power plants, as they are highly dependent on the weather conditions. The market should also be reasonably liquid and developed.

Another concern is the availability of free historical electricity price data for research purposes, which are usually sold by the market operators or other subjects to private companies for profit.

All of these conditions are satisfied by the German electricity market,

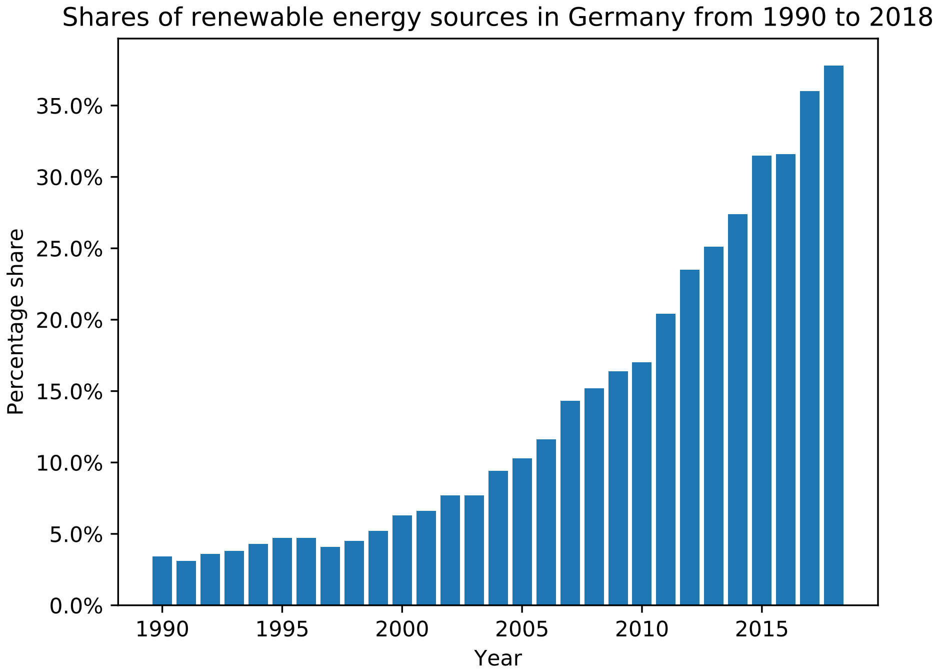


Figure 3.4: Shares of renewable energy sources in Germany [22]

which was eventually selected for the thesis. Germany has a stable electricity market, and for the past two decades, the share of renewable electricity sources was rising steadily up to 37.8% in 2018 [22], see Figure 3.4.

The main renewable electricity sources in Germany are the wind power plants with the production over 111 TWh in 2018 (both the on-shore and off-shore plants combined), and photovoltaic power plants with production over 46 TWh in the same year. In 2018, the total production from renewable electricity sources was around 225 TWh; thus these two sources made almost 70% of the renewable sources in Germany [22].

3.3 Data sources analysis

One of the goals of this thesis is to analyze the available electricity price and weather data. This includes the selection of appropriate data sources, their retrieval, preprocessing, and finally the analysis itself.

3.3.1 Electricity price data

The leading energy exchange in central Europe is the European Energy Exchange (EEX) [23]. It operates a market that includes energy and associated

Model	Spatial resolution [°]	Time resolution [h]	Number of forecast steps [h]
GFS [25]	0.25/0.5	3	384
ICON [26]	0.25	3	78
GEM [27]	0.24	3	240
NAVGEM [28]	0.5	3	180

Table 3.1: Popular weather forecast models comparison. The spatial resolution of GFS model is 0.25 degrees, but the archive data (older than 1 month) are only available in 0.5 degrees resolution.

commodities, such as power derivative contracts and emission allowances. According to 2017 annual report [24] (the latest available at the time of writing this thesis), the volume of traded electricity on the spot market amounted to 543 TWh.

The EEX group kindly provided free access to their historical market data for the purpose of writing this thesis.

3.3.2 Weather data

The weather forecasts are produced by numerical weather prediction (NWP) models. They compute weather predictions based on the initial state of the atmosphere at the base time and a large set of parameters. Calculating these models is computationally extremely demanding, and only a handful of global weather forecast models exist.

Most of the models run 2 or 4 times a day. The models are differently parameterized and feature a different number of forecast steps and different spatial and time resolution. They also forecast a different set of elements (such as temperature or cloud cover).

In Table 3.1, there is an overview of some of the frequently used models. One of the most popular of them is the GFS (Global Forecast System) model, which is run by the US National Weather Service (NWS). Because of the US legislation, the produced forecasts are available for free in the public domain. In 2016 comparison [29], GFS ranked 4th best among the global models.

For the purpose of this thesis, GFS is the best choice, because it is available for free in the public domain, but still provides reasonably accurate predictions. Considering the fact that the weather forecasts (which are 2D grids for each element) have to be transformed into single time series for each element, the slightly better performance of other models would probably not lead to significantly better results.

3.4 Modeling approaches

In this thesis, two different approaches to the electricity price forecasting are evaluated, namely the statistical model and the machine learning model. The selection of these models is described in Sections 3.4.1 and 3.4.2, respectively. To improve the forecasts, an algorithm for combining these two models is proposed in Section 3.4.3.

3.4.1 Statistical models

Since the electricity price is time series data with seasonality components, and there are exogenous explanatory variables - the weather data - a convenient model for the data is SARIMAX. It includes the simpler models, e.g., the ARIMA models, that were previously successfully exploited for the electricity price forecasting [17, 18, 19].

SARIMAX, being an extension of the basic ARIMA model, allows to include exogenous regressors in the prediction, which makes it possible to use the weather conditions in the forecasting. There is also an implementation of the model in `statsmodels` module for `python3` programming language, which makes it easy to evaluate.

3.4.2 Machine learning models

From the machine learning approaches, the artificial neural networks (ANNs) were selected as candidates for modeling. ANNs are a very popular tool for modeling a large variety of different problems. Many papers regarding using them to forecast the electricity prices were published, with good results regarding its accuracy [17, 21].

Also, the practical expertise of the author of this thesis suggests, that these models might be convenient for the electricity price forecasting.

3.4.3 Model averaging

In real-life electricity price forecasting, the performance of the chosen model may vary in different situations. For example, some simple models can perform very well when the prices obey the usual daily cycle, but they can fail to give reliable prediction in more difficult conditions. On the other hand, more complicated models might predict unusual price development well, but perform worse in other conditions.

Given the unstable nature of the electricity price, it is a good idea to create several different models, and *combine* their predictions in some way, to gain from their diversity. There are different techniques to achieve this goal. In this thesis, the aim is to use an approach that is consistent with the probability theory. It is proposed in the following subsection.

3.4.3.1 Probabilistic interpretation

Assume that we have a total of n models $M^{(1)}, \ldots, M^{(n)}$ that provide predictions $Y_{t+1:T}^{(i)} = \left(y_{t+1}^{(i)}, \ldots, y_{t+T}^{(i)}\right)$, where $i \in \{1, \ldots, n\}$ is the model index and $T \geq 1$ is the prediction horizon.

We want to find an optimal combination of predictions $Y_{t+1:T}^{(i)}$ that reflects the probabilities of $M^{(i)}$ being "correct". We can suppose that some of the models are better than the others, and therefore their predictions should be preferred over the rest of the models; however, it is not possible to identify them exactly on a finite dataset. Still, we can think of the vector of models $M = \left(M^{(1)}, \ldots, M^{(n)}\right)$ as a categorical variable with distribution

$$M \sim Cat\left(M^{(1)}, \ldots, M^{(n)}; \omega^{(1)}, \ldots, \omega^{(n)}\right), \tag{3.1}$$

where the vector $\omega = \left(\omega^{(1)}, \ldots, \omega^{(n)}\right)$ contains the probabilities of the models $M^{(1)}, \ldots, M^{(n)}$ being correct and Cat is the categorical distribution.

The probabilities ω have to be estimated from a finite dataset that is available for modeling. Using the Bayesian approach, the Dirichlet distribution can be chosen as the a priori distribution for the estimate of ω,

$$\omega \sim Dir\left(\omega^{(1)}, \ldots, \omega^{(n)}; \alpha^{(1)}, \ldots, \alpha^{(n)}\right), \tag{3.2}$$

where $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$ are the hyperparameters of the Dirichlet distribution. Using Bayes' theorem, the proper estimate of ω can be written as

$$\pi(\omega | y_t, \ldots, y_{t-\tau}, \theta) \propto f(M|\theta)\pi(\omega), \tag{3.3}$$

where f is the probability density function, θ is a parameter and $\pi(\omega)$ is the a priori distribution of ω. However, this estimate cannot be used because $f(M|\theta)$ is not available, and therefore has to be replaced by some other estimate. For probability models, we can use model likelihoods (previously defined in Section 1.1) to calculate the estimate $\widehat{\omega}$,

$$\widehat{\omega^{(i)}} = \frac{M^{(i)}\left(y_t, \ldots, y_{t-\tau} | \theta^{(i)}\right)}{\sum_{i=1}^{n} M^{(i)}\left(y_t, \ldots, y_{t-\tau} | \theta^{(i)}\right)}, \tag{3.4}$$

where $\theta^{(i)}$ are the parameters of ith model, the numerator in the fraction is the likelihood of last $\tau+1$ measurements and the denominator is the normalization term ensuring that $\sum_{i=1}^{n} \widehat{\omega^{(i)}} = 1$.

To include non-probability models (such as ANNs) into the averaging, the likelihoods cannot be used either, because they are not available for these models. Instead, we have to use another metric, for example, RMSE.

After estimating the vector $\widehat{\omega} = \left(\widehat{\omega^{(1)}}, \dots, \widehat{\omega^{(n)}}\right)$, the final prediction can be calculated as the weighted average of the individual models predictions using $\widehat{\omega}$ as the weights,

$$Y_{t+1:T} = \sum_{i=1}^{n} \widehat{\omega^{(i)}} Y_{t+1:T}^{(i)}. \tag{3.5}$$

Using this approach, we exploit the current modeling performance measure - the weights - as factors describing the expected future performance. The idea is inspired by the Bayesian model averaging [30], which adopts a purely probabilistic viewpoint based on the predictive performance of traditional statistical models.

Chapter **4**

Data

This chapter deals with the data that are used for the formulation of forecasting models. All stages of the data preparation are discussed, including the selection of proper data sources, data downloading and preprocessing, and analysis of the prepared data.

4.1 Data retrieval

In this section, the process of data retrieval is described. In the following sections, the retrieval of the electricity price data and the weather data is discussed separately. For the purposes of data downloading, several python scripts were created. They can be found in `data/data_downloading` folder.

4.1.1 Weather data

Weather forecasts from the GFS model are available for free in the public domain, and can be download from GFS archive [31] in `grib2` format. At the time of downloading the files, continuous forecasts were available from May 1, 2017, to the present date. Since large parts of forecasts before this date were not available, only data from the dates between May 2017 and December 2018 are used in the thesis.

There are 4 runs of the GFS model each day - at midnight, at 6:00 AM, at noon, and 6:00 PM. For the purpose of electricity price forecasting, the midnight run is the best choice, as the forecasts produced by it are available each day around 4:00 AM, which leaves enough time for generating the forecasts for the next day before the spot market closes (which is around noon). The forecasts are available in multiple `grib2` files. Each of the files contains the forecasts of all of the elements (e.g., temperature, cloud cover) for a given hour.

For the downloading of GFS forecasts, `python3` script `download_gfs.py` was prepared. It contains a simple CLI interface and allows to set the date of the model run to download and specify the forecast hours ahead of the base time. For the purpose of this thesis, forecast hours from +3 to +27 with a 3-hour step were downloaded for each day. It is due to the fact that the forecast hour +0 does not contain all of the needed elements, so hour +24 from the previous day has to be used instead. The forecast hour +27 was downloaded because it is used to interpolate the weather data for 1:00 AM and 2:00 AM.

4.1.2 Electricity price data

The EEX group provided free access to large portions of their market data. The data are available from mid-June of 2000; however, for the purpose of this thesis, it is not necessary to use all of the available data. In the long-term, the electricity price is more affected by other factors than the weather, for example by inflation. In the thesis, the analysis and modeling are done on data from dates between May 2017 and December 2018. The reason for the choice of this range is that both the weather and the electricity price data are continuously available in a single version for it, so no additional preprocessing to deal with different versions and missing weather data is needed.

The EEX group offers multiple kinds of data about the spot electricity prices in separate files. There are both detailed data from the electricity auction (individual biddings) and final market results. All of the data are available in `csv`, `xls` and `xml` format, for German, Austrian, Swiss, and French markets separately. For the purpose of this thesis, the market clearing price (MCP) of the German market is used.

For the downloading, `python3` script `download_electricity_data.py` was prepared. It contains a simple function that downloads all the data (filtering only needed rows) from a given date range. The data are saved in separate `csv` files for each day for later preprocessing.

4.2 Data preprocessing

In this section, data preprocessing procedures are described. Weather data and electricity price data are discussed separately in different subsections.

4.2.1 Weather data

The weather forecasts from the GFS model are available in `grib2` files, in separate files for each forecast hour, with each of the file containing all of the available elements for the whole Earth. For further use, it is necessary to convert the data from `grib2` format to `csv`, filter out unneeded elements and the locations outside of Germany. Since the GFS model offers forecasts with 3-hour step, but the electricity prices data are hourly, the weather data are interpolated to obtain finer resolution.

The data transformation from `grib2` format to `csv`, as well as the element and location filtering is done by `transform_gfs.py`. It uses the `pygrib` module [32], providing easy-to-use interface for accessing the data from `grib2` files. The script filters 17 different elements. The spatial coverage was selected to include the whole of Germany with 0.5° resolution, which is illustrated in Figures 4.1 and 4.2. The output of this script are `csv` files for each day and coordinate that contain the forecasts for the selected elements with 3-hour step.

The interpolating procedures for finer time resolution (1 hour) are contained in the `python3` script `interpolate_gfs.py`. Simple linear interpolation between two known time points is implemented there. For two known times t_0, t_3 and known forecasts F_0, F_3, interpolated values F_1 and F_2 can be computed using formulas

$$F_1 = F_0 + \frac{F_3 - F_0}{3} \tag{4.1}$$

and

$$F_2 = F_0 + 2 \cdot \frac{F_3 - F_0}{3}. \tag{4.2}$$

For the purpose of performing downloading and basic preprocessing simultaneously, the `bash` script `gfs_batch.sh` encapsulates all of the mentioned scripts. It takes one argument specifying the date and run of the GFS model to download and process.

At the time of downloading the weather data, several missing model runs in the used date range were revealed. These values were manually substituted by the most recent previous model run, which was usually 6:00 PM or 12:00 PM of the previous day.

In the previous steps, `csv` files with hourly forecasts for each point in Germany were prepared. However, there are separate time series for each coordinate in the used grid (intersections of parallels and meridians in Figures 4.1 and 4.2) for each element. For the analysis, it is necessary to convert them into a single time series for each of the elements. There are multiple ways to achieve this, the simplest one being simply averaging the data across the grid. However, this approach does not take into account the fact that the wind and photovoltaic power plants are not distributed equally across the country. Because of this, the more appropriate solution is to use a weighted average, where the weights reflect the amount of electrical capacity of the renewable power plants at the coordinates.

Data about renewable power plants are available from Open Power System Data [33]. At the time of downloading the data for this thesis, the newest available data were from March 8, 2018. The data contain GPS coordinates of solar and wind power plants and their electrical capacities in MW. To achieve the same spatial resolution as in the weather data (0.5°), the coordinates for each power plant are changed to the nearest point in 0.5°×0.5° grid.

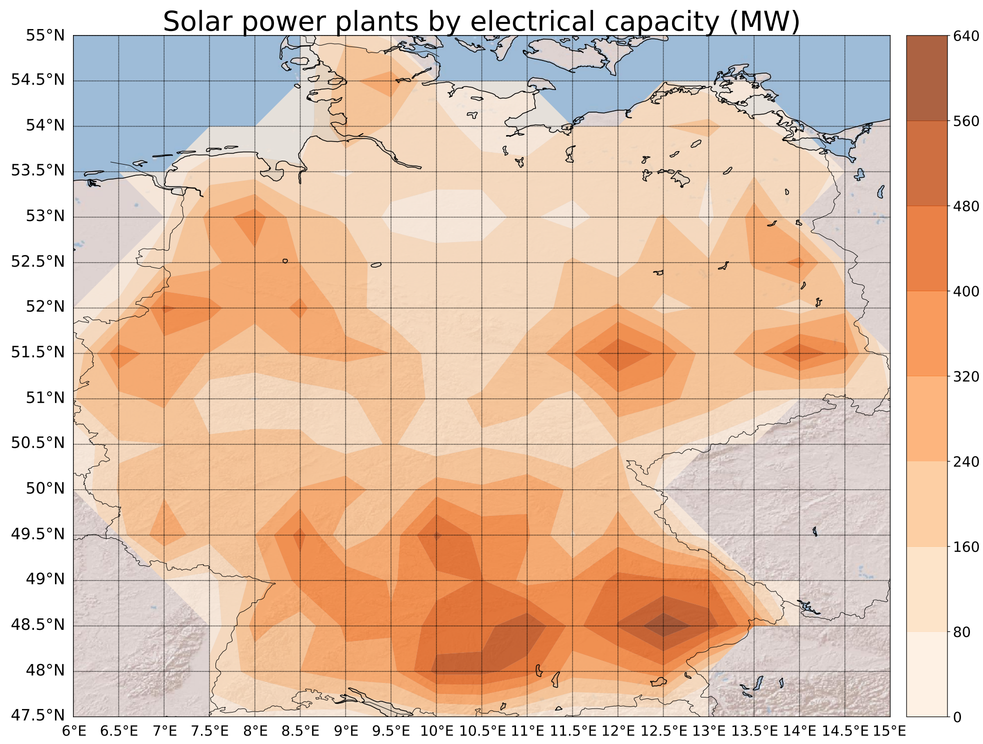


Figure 4.1: Solar power plants by electrical capacity

For each point in the grid, the sum of electrical capacities is subsequently computed (for both the solar and wind power plants separately). Resulting aggregated capacities for solar and wind power plants can be seen in Figures 4.1 and 4.2, respectively. These aggregated capacities are then used as weights in the weighted averaging procedure, which yields a single time series for each element. Forecasts for cloud cover, short-wave irradiation, and long-wave irradiation are weighted by aggregated capacities of solar power plants, whereas wind (decomposed into two vectors `U` and `V`, describing the westward and northward part, respectively) is weighted by the aggregated capacities of the wind power plants. The forecasts for temperature are weighted by the sum of both types of aggregated capacities.

Let N be the number of the points in the grid, $\boldsymbol{w} = (w_1, \ldots, w_N)$ vector of weights and F_p forecast for given element in given time in point p. Weighted average F of the prediction is then computed using formula

$$F = \frac{\sum_{p=1}^{N} F_p \cdot w_p}{\sum_{p=1}^{N} w_p}. \tag{4.3}$$

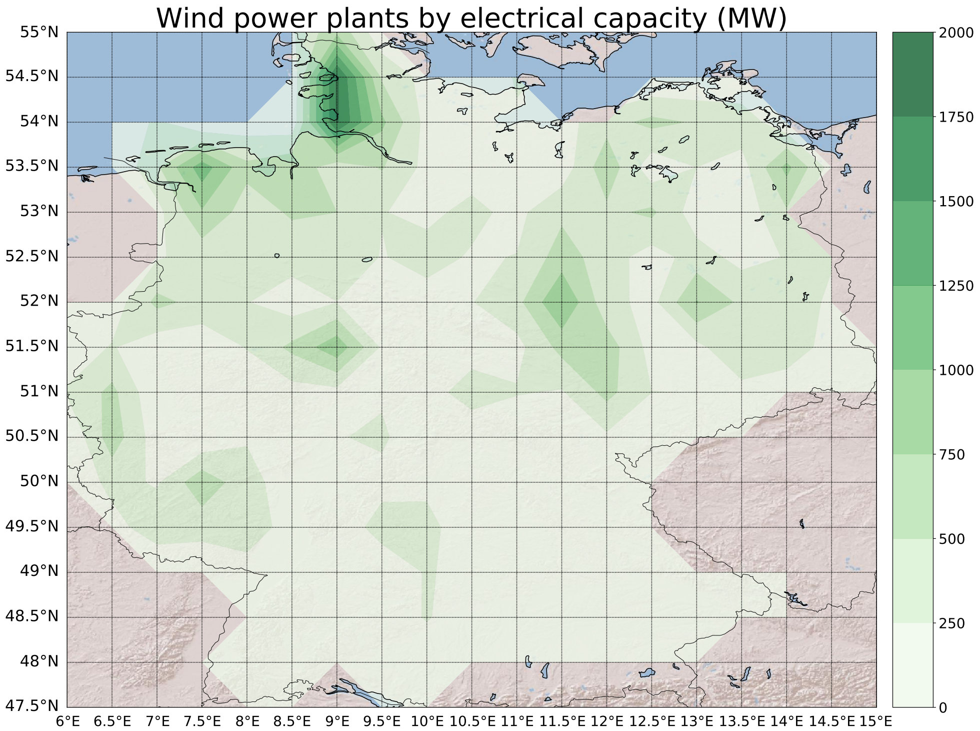


Figure 4.2: Wind power plants by electrical capacity

4.2.1.1 Weather elements

For the purpose of forecasting, 7 elements that are most likely to affect the electricity prices were selected. These elements are briefly described in the following list. In the parentheses, abbreviated names of the elements that are used in the datasets are stated.

- Temperature (`temperature` or `temp`)

  The GFS model uses Kelvin (K) units for temperature; however, in the preprocessing, the temperature is converted to degree Celsius (°C). The model provides forecasts for multiple vertical levels, but for the purpose of this thesis, only 2 meters above the ground level is used.

- Cloud cover (`clouds`)

  The cloud cover forecasts are available for low (up to 2km above the ground), middle (2–7km above the ground), high cloud cover (5–13km above the ground), and an aggregation of all the levels (which is used in the thesis). The values in the data denote the percentage of sky that is covered with clouds, considering all of the vertical levels. The cloud cover directly affects the production of photovoltaic power plants.

- Long-wave irradiation, Short-wave irradiation (`lw`, `sw`)

  These elements describe the amount of long-wave (wavelength between 3 µm and 100 µm) and short-wave radiation (wavelength between 0.2 µm and 3 µm) that reaches the earth surface, measured in $\mathrm{W\,m^{-2}}$. Along with the cloud cover, they affect the electricity production of solar power plants.

- U wind component, V wind component (`uw10`, `vw10`)

  Wind forecasts are provided decomposed into `U` and `V` vectors of the wind, measured in $\mathrm{m\,s^{-1}}$. The U component describes the part of the wind that blows from the east to the west, whereas the V component describes the wind blowing from the south to the north. The values are negative when the wind blows in the other direction (from the west to the east and from the north to the south).

- Wind speed (`wind_spd`)

  The wind speed directly affects the production of wind power plants, a negative correlation with the electricity price is expected. It is not directly forecasted by the GFS model, but it can be easily computed from the `U` and `V` components, using formula

$$W_{spd} = \sqrt{W_u^2 + W_v^2}. \tag{4.4}$$

4.2.2 Electricity price data

Weather forecasts from the GFS model use Coordinated Universal Time (`UTC`) time standard, which does not adopt the daylight saving time. On the other hand, the EEX Group expresses the time in Central European (Summer) Time (C(E)ST), which adopts the daylight saving time. Because of this, the time that EEX uses is 1 hour ahead of `UTC` time in winter and 2 hours ahead in summer. To synchronize the two times, the time of the electricity price dataset is converted to the `UTC` time standard.

4.3 Data analysis

In this section, an analysis of the preprocessed data is discussed. Weather data and electricity price data are dealt with separately in different sections. For the purpose of analysis and forecasting, several new features are created. These include separated parts of date and time (hour, day of the week, month) and an indicator feature `is_weekend`, which is set to `1` on Saturday and Sunday.

The analysis is carried out using `python3` programming language and external modules. These include `pandas` [34] for the data manipulation and analysis, `numpy` [35] for various computing tasks and `matplotlib` [36] for plotting.

	temperature	clouds	lw	sw	uw10	vw10	wind_spd
mean	11.3	56	310.3	161.3	1.1	0.6	3.5
std	7.3	26.6	37.2	204.1	3.1	2.2	1.9
min	-10.2	0	177.5	0	-9.6	-8.3	0.02
max	31.9	100	406.4	778.3	11.9	8.7	12.9
Q_5	-0.3	9.7	241.2	0	-4	-2.9	0.9
Q_{10}	1.5	18.4	260.1	0	-3	-2.2	1.3
Q_{25}	5.3	35.4	288.3	0.6	-1	-1	2.2
Q_{50}	11.7	57.7	313.1	54	1.2	0.5	3.2
Q_{75}	16.8	78.5	337.1	285.2	3.3	2.1	4.7
Q_{90}	20.9	90.8	356.3	514.6	5.1	3.4	6.2
Q_{95}	23.1	95.4	365.5	597.5	6.1	4.3	7.1

Table 4.1: Statistical properties of the weather data

4.3.1 Weather data

The Table 4.1 summarizes selected descriptive statistics of the transformed dataset. All of the variables are in expected bounds. Further description of the variables can be found in Section 4.2.1.1.

4.3.2 Electricity price data

Firstly, some basic statistical properties of the prices data should be evaluated. All the values are electricity prices in EUR/MWh on the spot market. The dataset contains 14641 records, with mean value and standard deviation being 39.66 and 18.18, respectively. Minimal recorded value in dataset is -83.6, maximal is 129.56. In Table 4.2, selected percentiles of the data are stated. Histogram 4.3, depicts the distribution of the data. From both the table and histogram, it can be seen that prices under 11 EUR/MWh and over 70 EUR/MWh are very rare.

As mentioned in the introduction, the electricity price tends to be very volatile. In Figure 4.4, the electricity price between May 2017 and December 2018 is depicted. No obvious trend or seasonality can be seen in this large scale graph. However, very clear daily seasonality is apparent from Figure 3.2. The same seasonality can be observed in Figure 4.5, which represents hourly means of the electricity prices in 2018. From these graphs, it is obvious that the daily price tends to have two maxima - in the morning and the evening, and two minima - in the afternoon and in the night. However, the typical daily cycle can be disrupted, so this behavior does not always occur, as can be seen in Figure 3.3.

From the hourly mean graph in Figure 4.5, it is also possible to see how the prices change through the seasons of the year. Except for December (winter), the price curve is very similar, just shifted upwards or downwards.

	value
mean	39.66
std	18.18
min	-83.6
max	129.56
Q_5	11.21
Q_{10}	19.82
Q_{25}	30.02
Q_{50}	39.12
Q_{75}	49.94
Q_{90}	61.79
Q_{95}	69.59

Table 4.2: Statistical properties of the electricity prices data

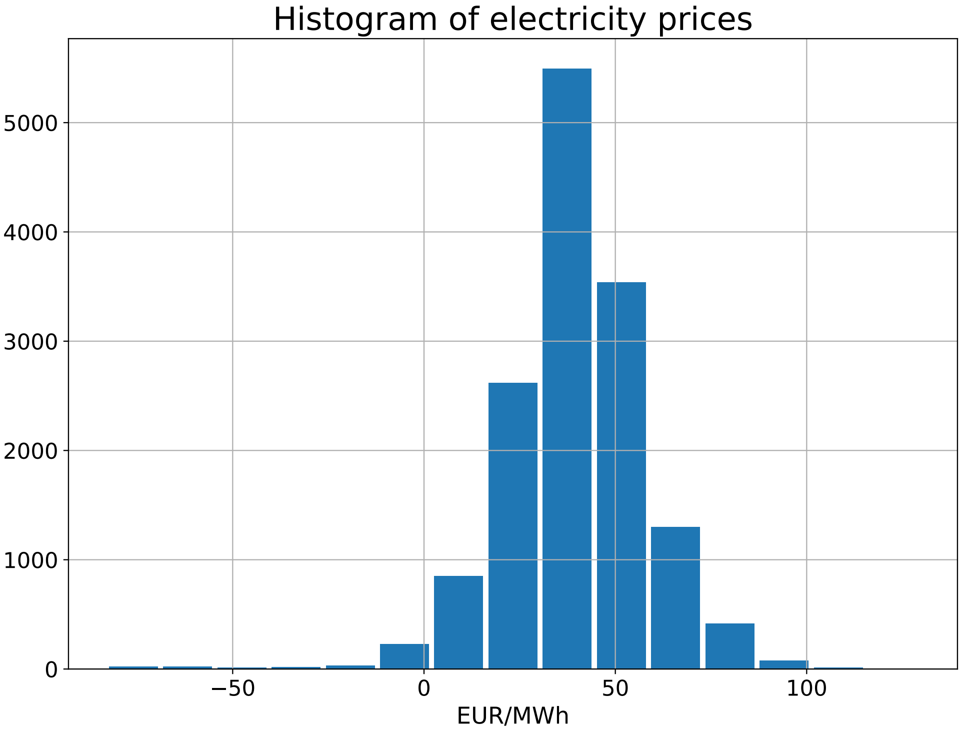


Figure 4.3: Histogram of the electricity prices

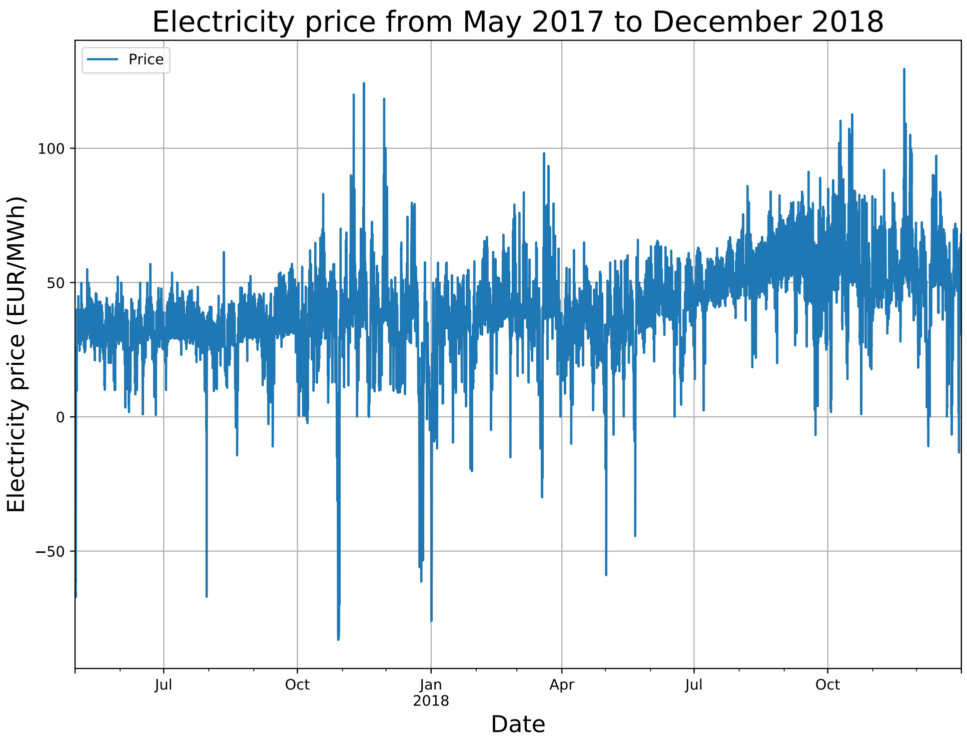


Figure 4.4: The electricity price from May 2017 to December 2018

In December, the afternoon minimum is less distinctive, which might be caused by the necessity of heating throughout the day.

The electricity price time series also has clear weekly seasonality. Mean prices by days of the week in the dataset are shown in Figure 4.6. This seasonality is most likely caused by decreased electricity consumption during weekends which can be explained by the fact that most factories and other large power consumers do not operate on Saturday and Sunday.

As mentioned in Section 3.1 and shown in Figure 3.3, the electricity price can be negative. It does not happen very often though, negative prices occurred 241 times only, which makes 1.6% of the dataset. In Figure 4.7, histograms of the negative values grouped by `Day`, `Hour`, `Month` and `is_weekend` features can be seen. From the histogram, it is clear that this anomaly mostly happens at night, around noon, and on Sunday. On the other hand, it rarely occurs in the late afternoon, which is in accordance with the assumption that it should occur mostly in off-peak hours. Considering a relatively small number of negative prices, they can be discarded as outliers and replaced by the mean value of prices in the same hour, day in a week, and month from the dataset.

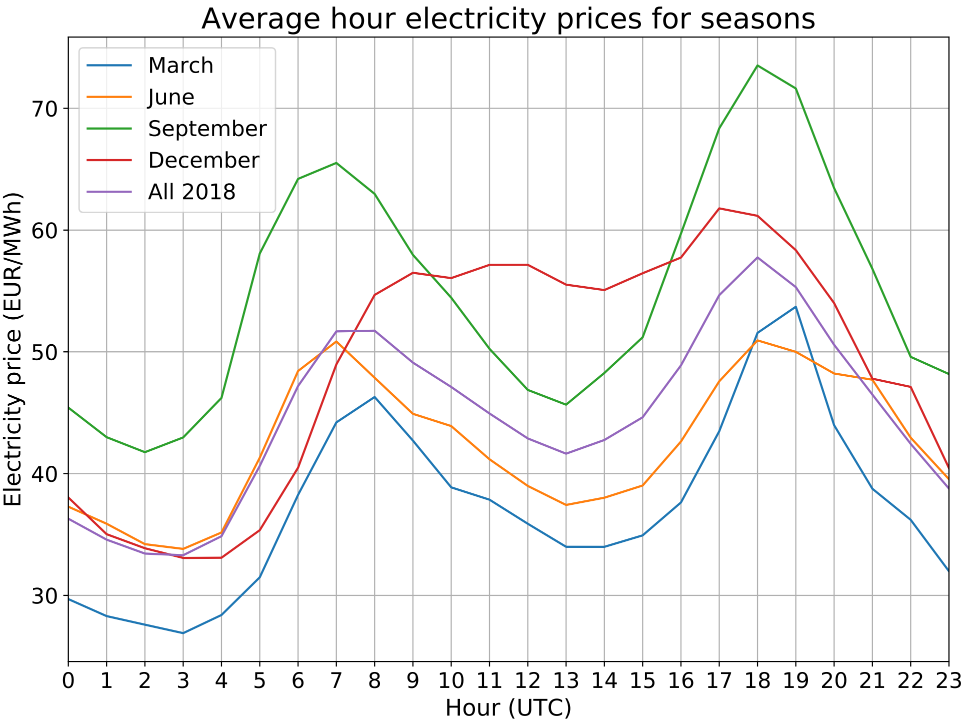


Figure 4.5: Mean electricity price by hours and seasons

4.3.3 Correlation between weather and electricity price

The basic overview of the relationship between the weather conditions and the electricity price can be obtained by calculating Pearson correlation coefficients. Values of the correlation coefficients between the variables are stated in Table 4.3.

From these values, it is obvious that wind has the most influence on the electricity price. Especially the feature `wind_spd`, which represents the absolute wind speed. The correlation is negative, which is in compliance with the assumption - we suppose that the stronger wind, the more electricity is generated by the wind power plants, and therefore the lower the market price would be.

The cloud cover variable has much lower correlation with the electricity price than the wind. Surprisingly, the correlation is also negative, which is against the intuition (even though the correlation is not very strong). The assumption was that the correlation should be positive - the less clouded sky means more production of the solar power plants, and therefore the electricity price should be lower. The counterintuitive correlation can be caused by many factors. One of the most probable causes is the fact that the cloud cover feature is also included during the night when the production of solar power

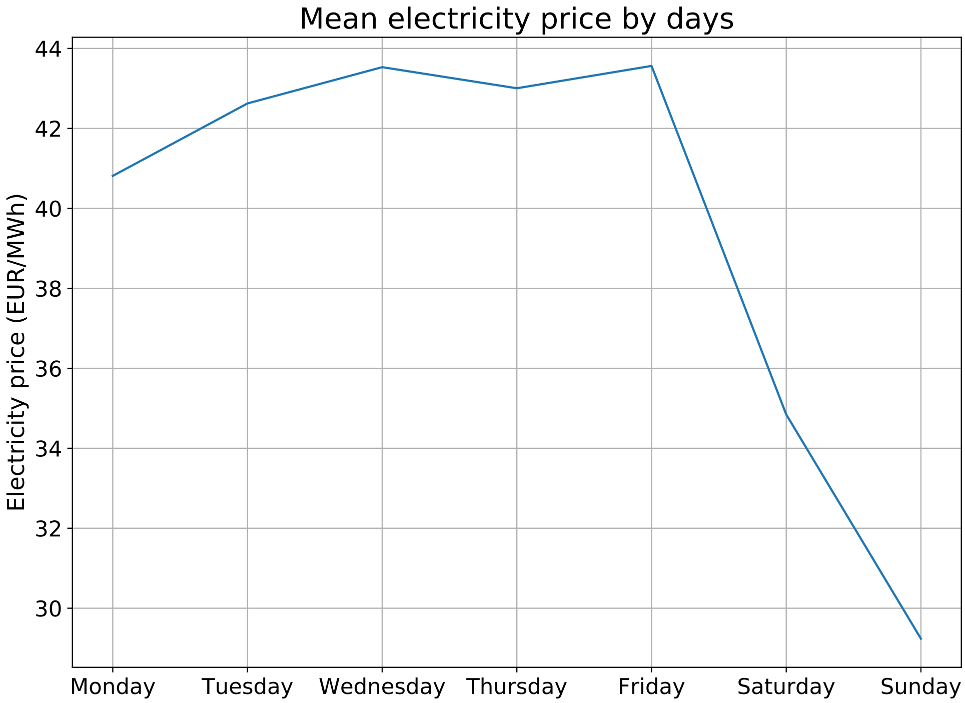


Figure 4.6: Mean electricity price by days

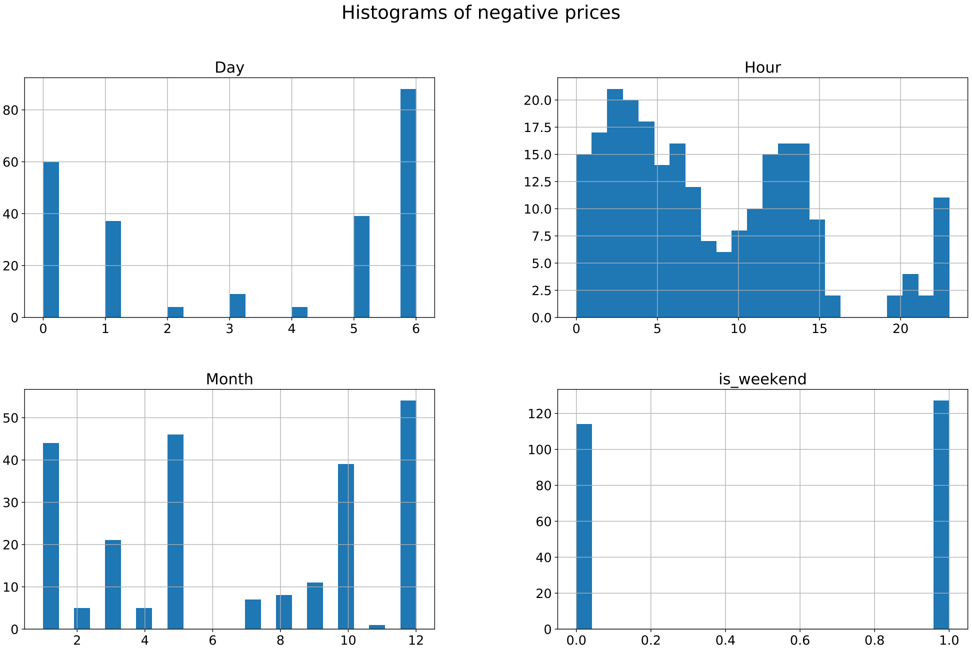


Figure 4.7: Grouped histograms of negative prices

	Price	**temp**	**clouds**	**lw**	**sw**	**uw10**	**vw10**	**wind_spd**
Price	1.00	0.06	-0.13	0.02	0.04	-0.24	-0.12	-0.45
temp	0.06	1.00	-0.27	0.87	0.64	0.06	-0.10	-0.22
clouds	-0.13	-0.27	1.00	0.09	-0.25	0.23	0.10	0.28
lw	0.02	0.87	0.09	1.00	0.40	0.20	-0.13	-0.13
sw	0.04	0.64	-0.25	0.40	1.00	-0.04	-0.18	-0.04
uw10	-0.24	0.06	0.23	0.20	-0.04	1.00	0.12	0.41
vw10	-0.12	-0.10	0.10	-0.13	-0.18	0.12	1.00	0.20
wind_spd	-0.45	-0.22	0.28	-0.13	-0.04	0.41	0.20	1.00

Table 4.3: Values of the Pearson correlation coefficient between the features

plants is always zero. Also, the spatial resolution of the weather data is 0.5°, which makes about 55.6 km in Germany. The solar power plants production is affected by the instantaneous cloud cover of the exact location of the power plant, which is generally very hard to predict precisely.

All of the other features have insignificant correlations with the price. However, the Pearson correlation coefficient measures the strength of the linear relationship, and it is possible that these features affect the electricity price nonlinearly.

Chapter 5

Modeling

In this chapter, the construction of predictive models for the electricity price forecasting is described. For predicting the prices, two different approaches were chosen to be implemented - a statistical model and a machine learning model. In Section 3.4, SARIMAX was proposed to be implemented as the statistical model and ANN as the machine learning model. To improve the forecasts, an implementation of an algorithm for combining the forecast of different models is proposed in Section 5.3, based on the method described in Section 3.4.3.

All of the modeling was implemented using `python3` programming language and external modules that implement the model fitting and forecasting and provide a high-level interface for these procedures. These modules are mentioned in the following sections.

5.1 SARIMAX models

The SARIMAX models were implemented using the `statsmodels` module [12]. It contains high-level methods for fitting the model and creating the predictions, and it has many hyperparameters that allow very detailed tuning of the models.

5.1.1 Parameter tuning

For SARIMAX models, the most important parameters to select are the orders p, q, d, P, Q, D and the seasonality s, which are described in Section 1.10.4. Simultaneously, the subset of the available exogenous variables to be included in the model has to be specified. Unfortunately, it is not possible to perform an exhaustive grid search of all the possible combinations of these parameters, because the fitting is computationally very demanding; thus only a subset of them is evaluated.

In order to keep the models simple and prevent overfitting, it is not possible to use the entire dataset when fitting the predictive model, but only include the most recent data (of some length). By evaluating the performance of the models (in terms of RMSE) for lengths 30, 60, 90, and 120 days, the ideal length of the training dataset was found out to be 60 days.

The selection of parameter s, the length of the seasonality, is fairly straightforward. Because the dataset contains hourly data and the electricity price has strong daily seasonality, the obvious value of s is 24.

In the data preprocessing, a total of 11 exogenous regressors that can be used in the model have been prepared. These variables are `Hour`, `Month`, `Day`, `is_weekend`, `temperature`, `clouds`, `lw`, `sw`, `uw10`, `vw10`, and `wind_spd`. For a set that contains n elements, 2^n possible subsets exist, so in this case, an exhaustive search of all the possible subsets would require evaluating $2^{11} = 2048$ combinations, which would be computationally very demanding. Instead, 6 different subsets of exogenous regressors were selected for evaluation. These subsets are:

- No exogenous regressors
- `wind_spd`
- `wind_spd`, `clouds`
- `wind_spd`, `clouds`, `lw`, `sw`
- `is_weekend`, `temperature`, `clouds`, `lw`, `sw`, `uw10`, `vw10`, `wind_spd`
- `Hour`, `Month`, `Day`, `is_weekend`, `temperature`, `clouds`, `lw`, `sw`, `uw10`, `vw10`, `wind_spd`

The parameters p, q, d, P, Q, D define the complexity of the model. The models should be kept reasonably simple, in order to prevent overfitting. The following inequalities can be used as a rule of thumb for setting the parameters [37],

- $P + Q \leq 2$,
- $D + d \leq 2$,
- $p + q \leq 3$.

Using these constraints, a subset of all the possible combinations of all the above parameters was used to fit the models. The models were then compared using AIC after fitting the models on randomly selected 60 days. A total of 6 different SARIMAX models with a low value of AIC were selected to be used for the predictions. Their parameters and achieved AIC are stated in Table 5.1.

Model no.	Exogenous regressors	p	d	q	P	D	Q	AIC
1	is_weekend, temp, clouds, lw, sw, uw10, vw10, wind_spd	1	0	1	0	1	1	8096.5
2	is_weekend, temp, clouds, lw, sw, uw10, vw10, wind_spd	1	0	2	0	1	1	7904.94
3	is_weekend, temp, clouds, lw, sw, uw10, vw10, wind_spd	2	0	1	0	1	1	7911.31
4	wind_spd	2	0	0	0	1	1	7942.45
5	is_weekend, temp, clouds, lw, sw, uw10, vw10, wind_spd	2	0	0	0	1	1	7914.93
6	is_weekend, temp, clouds, lw, sw, uw10, vw10, wind_spd	0	0	2	0	1	1	8026.7

Table 5.1: Selected SARIMAX models

Given the fact that the model averaging algorithm is later used to combine the forecasts, the performance of a particular model is not critical because the algorithm should prefer the models that perform the best.

5.1.2 Implementation

The fitting of the SARIMAX model and obtaining the predictions is implemented in class `SarimaxModel` in `models.py`. It contains 3 member functions `fit`, `predict`, and `predict_on_train`.

The member function `fit` encapsulates the model fitting. It takes two arguments, `start` and `end`, which describe the start and the end of the training data in `YYYY-MM-DD` format.

The function `predict` encapsulates the predicting routines. The predictions are always created for the next day of the last day of the training data and contain 24 prediction steps.

The last member function `predict_on_train` encapsulates additional fitting and predicting that is needed for the model averaging algorithm and is described in Section 5.3.

5.2 Neural network models

Neural networks models were implemented using `scikit`'s implementation `MLPRegressor` [38]. It is an implementation of multi-layer perceptron regressor that allows setting many different parameters. Most important of them is the architecture of the network - the number of hidden layers and the number of neurons in them.

5.2.1 Adding lags to dataset

To accommodate the neural network to process time series data, it is necessary to add past values of the target variable to the dataset. However, it is not possible to directly add the previous value of the target variable, because it is not available in real forecasting, due to the fact that the forecasts are done for 24 steps ahead. However, it is possible to add feature `day_ago`, which contains the electricity price from 24 hours ago.

To reflect the daily and weekly seasonality of the electricity price, 4 new features were added - `day_ago`, `2_days_ago`, `week_ago`, and `2_weeks_ago` that represent lags 24, 48, 168 and 336, respectively. Adding more lags could potentially lead to overfitting and therefore is not done.

5.2.2 Parameter tuning

The implementation of ANN used in the modeling offers many parameters to be tuned. It is not possible to tune all of them, as it would take too long. Instead, 3 of these parameters that were supposed to have the most influence were selected to be tuned, along with the length of the training data.

The most important parameter to tune for the neural network is the architecture - the number of hidden layers and the number of neurons in them. The model should be kept reasonably simple, as it is prone to overfitting with many hidden layers and neurons. A total of 9 different architectures were selected to be evaluated - single hidden layer with 2, 5 and 10 neurons, two hidden layers, each having 2, 5 or 10 neurons and three hidden layers, with 2, 5 and 10 neurons in each.

Another parameter that was selected to be tuned was `learning_rate`. It defines the learning rate scheduling for weight updates [38]. The available settings are:

- `constant` The learning rate is kept constant; it is the default setting.
- `invscaling` Gradually decreases the learning rate.
- `adaptive` Keeps the learning rate constant, as long as the training loss keeps decreasing. When two consecutive epochs fail to decrease the training loss enough, the learning rate is decreased.

Architecture	learning_rate	early_stopping	Training dataset length	RMSE
(5,5)	constant	False	120 days	7.75
(10,10)	constant	False	120 days	7.54

Table 5.2: Selected ANN models

The last parameter that was tuned was the early_stopping parameter, which sets whether to stop the training early when the validation score is not improving [38]. If set to true, it sets aside 10% of training data as validation and terminates the training when the validation score is not improving enough for a set number of consecutive epochs.

The last option that was tuned was the length of the training dataset. Four different lengths were evaluated - 30, 60, 90 and 120 days.

To improve the performance of the model, data scaling prior to the training was also tried, but discarded, because the models performed worse than with the original data.

For the parameter tuning, 92 randomly selected days from the dataset were chosen, and the models were trained to yield predictions for two consecutive days with separate training for each of the two days, which made up a total of 184 days. For each of the parameter combinations, the average RMSE was calculated, and the models with the lowest values were selected. The selected models along with their average RMSE are stated in Table 5.2.

5.2.3 Implementation

For the purpose of fitting the ANN model and creating the predictions, class `NeuralNetworkModel` in `models.py` was implemented. It contains identical member functions as the `SarimaxModel` class (described in Section 5.1.2) and therefore it is not be discussed here again.

5.3 Model averaging algorithm

As mentioned in Section 3.4.3, for the purpose of predicting the electricity price, it is a good idea to combine several models to achieve more accurate and robust forecasts. To combine the models, the following algorithm was proposed and implemented:

1. Fit models $M_*^{(1)}, \dots, M_*^{(n)}$ on data $X_{t-\tau:t-\upsilon}$, where $\tau > 1$ is the length of the train dataset and $\upsilon < \tau$ is the length of predictions on which the weights of the models are estimated on.

2. Using models $M_*^{(1)}, \dots, M_*^{(n)}$, create predictions $Y_{t-\upsilon+1:t}^{(i)}$.

3. Using numerical optimizer, estimate weights $\widehat{\omega} = \widehat{\omega^{(1)}}, \dots, \widehat{\omega^{(n)}}$ that minimize RMSE of the ensemble prediction $Y_{t-\upsilon+1:t}$,

$$Y_{t-\upsilon+1:t} = \sum_{i=1}^{n} \widehat{\omega^{(i)}} Y^{(i)}_{t-\upsilon+1:t}, \tag{5.1}$$

and satisfy conditions

$$\sum_{i=1}^{n} \widehat{\omega^{(i)}} = 1, \tag{5.2}$$

$$\widehat{\omega^{(i)}} \geq 0, i \in \{1, \dots, n\}. \tag{5.3}$$

4. Fit models $M^{(1)}, \dots, M^{(n)}$ on data $X_{t-\tau:t}$.

5. Using models $M^{(1)}, \dots, M^{(n)}$, create predictions $Y^{(i)}_{t+1:T}$.

6. Calculate ensemble prediction $Y_{t+1:T} = \sum_{i=1}^{n} \widehat{\omega^{(i)}} Y^{(i)}_{t+1:T}$.

In step 1, the models are not trained on full training dataset, but a part of it (last υ measurements) is excluded and the temporary models $M^{(1)}_{*}, \dots, M^{(n)}_{*}$ are trained only on data older than τ. The temporary models then create predictions $Y^{(i)}_{t-\upsilon+1:t}, i \in \{1, \dots, n\}$ (step 2), which are used to estimate the optimal weights for the ensemble model (step 3). Without this step, estimating the weights to minimize the RMSE of the models on the training dataset led to a very strong preference of the SARIMAX models over ANN, because they tend to have a better fit to training data, even though the average forecasting performance is very similar. This additional step ensures that the forecasting performance is comparable, which was experimentally proven to lead to better predictive results.

When the weights are estimated, the models are fitted on full training dataset (step 4), and they yield predictions for T timesteps ahead (step 5). The weights $\widehat{\omega}$ estimated in step 3 are used to calculate the final ensemble prediction (step 6).

5.3.1 Implementation

The model averaging algorithm described in previous Section 5.3 is implemented in class `ModelPool` in file `models.py`. It contains multiple member functions; the most important of them are discussed in the following list.

- `objective`

  This function returns the RMSE of the combined forecasts of the models, using weights provided as its argument. It is used as the objective for the numerical optimization of the weights to minimize the RMSE of the ensemble prediction in step 3.

- `optimize_weights`

  This function fits all of the models on data $X_{t-\tau:t-\upsilon}$ (step 1 from the averaging algorithm description), where υ is defined in constructor parameter `holdout_len` in `SarimaxModel` and `NeuralNetworkModel` classes. This fitting is implemented in `predict_on_train` member function in these two classes, which also yields predictions $Y^{(i)}_{t-\upsilon+1:t}$ (step 2).

  After the predictions are generated, the function uses an implementation of a numerical optimizer from the `scipy` library [39] to estimate the weights $\widehat{\omega} = \left(\widehat{\omega^{(1)}}, \dots, \widehat{\omega^{(n)}}\right)$ (step 3 of the algorithm) that satisfy conditions in Equations (5.2) and (5.3). The initial values of each weight estimate $\widehat{\omega^{(i)}}$ is set to $\frac{1}{n}$, where n is the number of the models.

- `predict`

  This function calculates the final ensemble predictions. First, it calculates predictions of each of the models and then it uses the estimated weights to create the final prediction. It should be used in `predict_long` function only because it requires the weights of the models to be set and models to be fitted.

- `predict_long`

  This function encapsulates the long-term predicting. It takes two arguments, `from_date` and `to_date`, which set the date range of the predictions (in `YYYY-MM-DD` format). For each of the dates in the range, all of the models are fitted to make predictions for the given date. Then the function `optimize_weights` estimates the weights of the models for the subsequent averaging. Then, `predict` function creates the final predictions using the estimated weights.

- `plot`

  For the purpose of visualizing the forecasts, the member function `plot` was implemented. It creates a plot using `matplotlib` library that contains the predictions from all of the models, the ensemble prediction and the real value of the target variable.

Chapter 6

Results

For evaluating the predictive performance of the models, predictions for 50 random days were created using all of the models and also the ensemble model, calculated by the model averaging algorithm. For each of the predictions, RMSE value was calculated, and then the average of these values was taken.

In Table 6.1, the measured average RMSE values for the models are stated. The `Parameters` column defines $(p, d, q)(P, D, Q)_s$ for the SARIMAX models, number of neurons in hidden layers (layers are separated by comma) for the ANN models, and the value of υ in days for the ensemble models. The best achieved RMSE for each model category is in **bold**.

From Table 6.1 it clearly follows that on average, all of the ensemble models performed better than any of the individual models. The best value of υ is 1 day. In average, both artificial neural network models performed worse than any of the SARIMAX models. However, for some days, ANNs were significantly better.

In many days, the ensemble predictions successfully assigned the largest weights to the best performing models, and therefore the final ensemble prediction achieved RMSE similar to the best-performing model. In some days, the ensemble prediction was even significantly better than the best performing model. For example, the ensemble forecast for 16 December 2018 performed almost twice better than the average of all the individual models, and significantly better than the best one. The plot for this date can be seen in Figure 6.1.

During some days, the electricity price experienced anomalous behavior, which none of the models could capture. It was most likely caused by some factors that cannot be successfully predicted. An example of this behavior can be seen in Figure 6.2, where the predictions for 21 May 2018 are visualized.

In some cases, the averaging algorithm assigned large weights to models that turned out to perform poorly. An example of this situation can be seen in Figure 6.3, where the only models that were assigned non-zero weights were the worst performing two. However, these situations are rare, as can be seen

Model number	Model	Parameters	Training length	Exogenous regressors	RMSE
1	SARIMAX	$(1,0,1)(0,1,1)_{24}$	60	is_weekend, temp, clouds, wind_spd, lw, sw, uw10, vw10	7.818
2	SARIMAX	$(1,0,2)(0,1,1)_{24}$	60	is_weekend, temp, clouds, wind_spd, lw, sw, uw10, vw10	7.821
3	SARIMAX	$(2,0,1)(0,1,1)_{24}$	60	is_weekend, temp, clouds, wind_spd, lw, sw, uw10, vw10	7.9
4	SARIMAX	$(2,0,1)(0,1,1)_{24}$	60	wind_spd	8.023
5	SARIMAX	$(2,0,0)(0,1,1)_{24}$	60	is_weekend, temp, clouds, wind_spd, lw, sw, uw10, vw10	7.888
6	SARIMAX	$(0,0,2)(0,1,1)_{24}$	60	is_weekend, temp, clouds, wind_spd, lw, sw, uw10, vw10	**7.721**
7	ANN	(10,10)	120	-	**8.383**
8	ANN	(5,5)	120	-	9.093
-	ENSEMBLE	1	-	-	**7.455**
-	ENSEMBLE	3	-	-	7.657
-	ENSEMBLE	5	-	-	7.874
-	ENSEMBLE	7	-	-	7.673

Table 6.1: Models performance. The *Parameters* column defines $(p,d,q)(P,D,Q)_s$ for SARIMAX models, number of neurons in hidden layers (layers are separated by comma) for ANN models, and the value of υ in days for the ensemble models.

from the average RMSE of the averaged models.

In the ensemble predictions (value υ set to 1 day), the model averaging algorithm tended to assign non-zero weights only to a small number of models. Non-zero weights were assigned in 34% of the cases only to 1 model, in 38% of the cases to 2 models, in 20% of the cases to 3 models and 8% of the cases to 4 different models.

In Table 6.2 and Figure 6.4, information about how often the models were assigned non-zero weights can be seen.

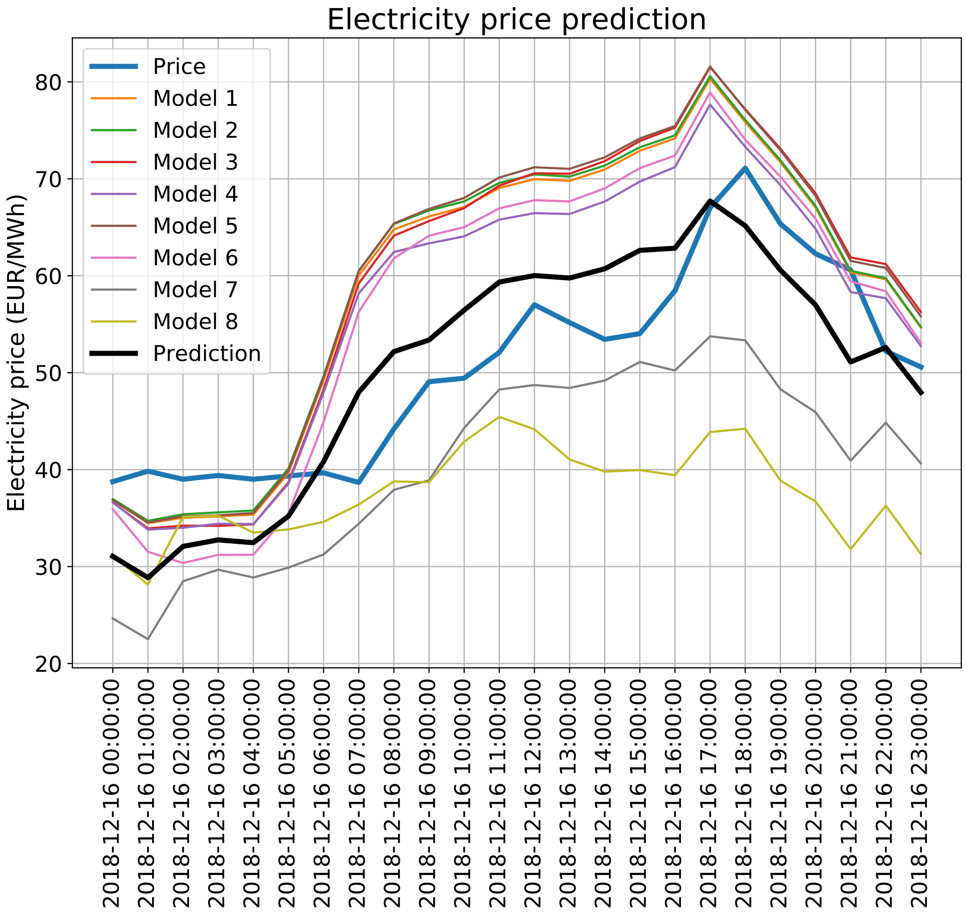


Figure 6.1: Example of good ensemble prediction. The ensemble prediction is labeled “Prediction”. In this case, the RMSE of the ensemble prediction is 6.34, while the best of the models achieved RMSE 10.22 and the average RMSE of the models was 12.24. In this case, model 2 was assigned weight 0.52 and model 7 had weight 0.48. The value υ was set to 1 day.

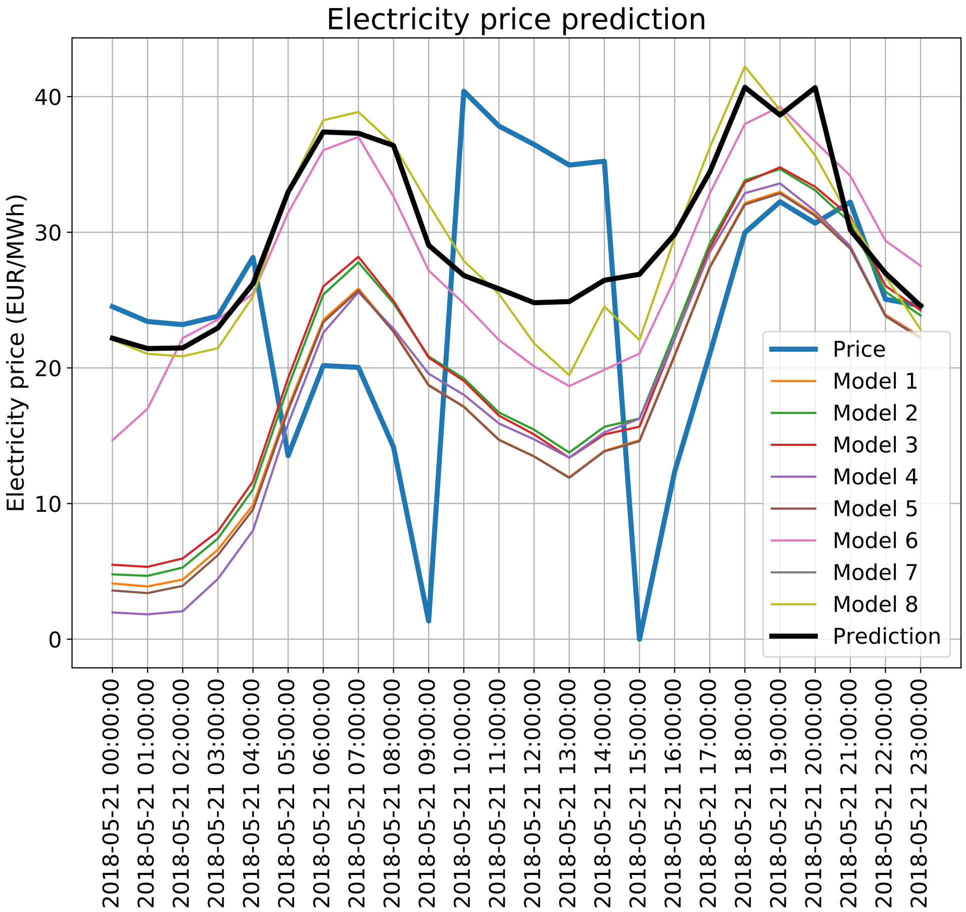


Figure 6.2: Example of uncommon prices development. The ensemble prediction is labeled “Prediction”. On this day, the electricity price experienced anomalous behavior, which was not captured by any of the models. It was most likely caused by some unpredictable factors. The value υ was set to 1 day.

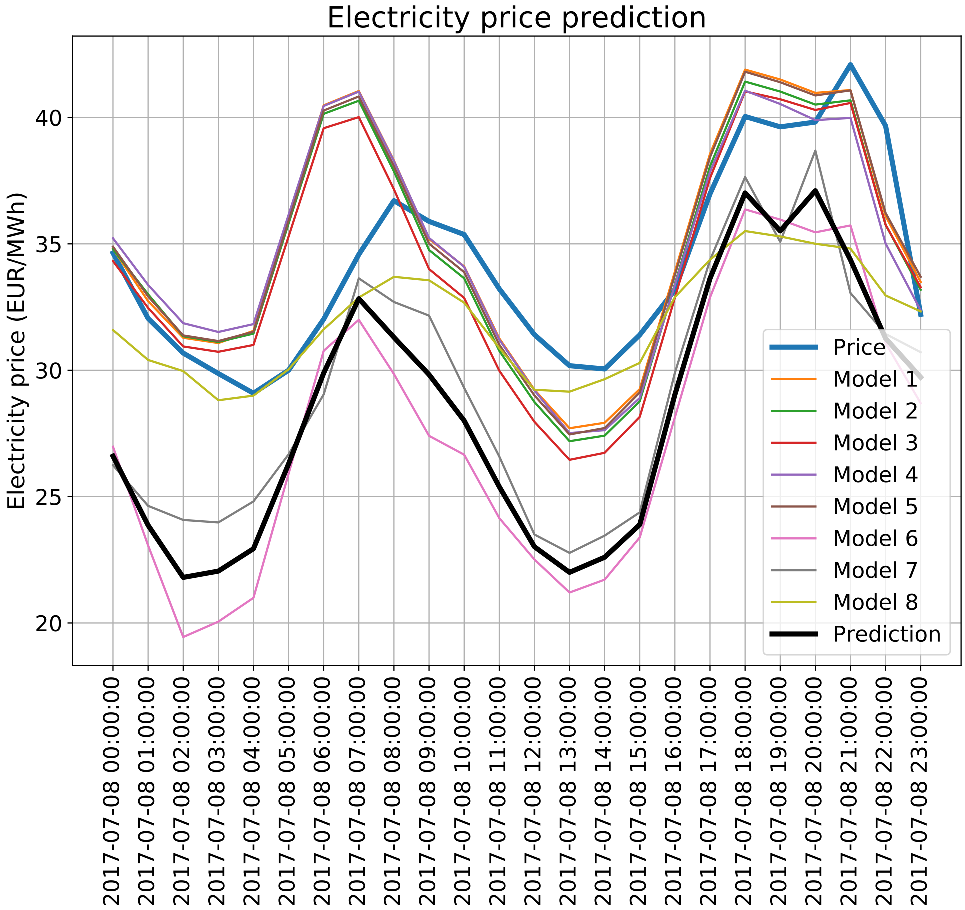


Figure 6.3: Example of bad ensemble prediction. The ensemble prediction is labeled “Prediction”. In this case, the averaging algorithm incorrectly assigned large weights to the models that performed the worst. It assigned weight 0.49 to model 6 and 0.51 to model 7. The ensemble prediction achieved RMSE 6.32, while an average of the models that had weight 0 was 2.99. The value υ was set to 1 day.

Model number	Number of non-zero weights	Percentage of non-zero weights
1	6	12%
2	5	10%
3	10	20%
4	18	36%
5	1	2%
6	13	26%
7	26	52%
8	22	44%

Table 6.2: The number of non-zero weights for models. The values represent the number of times, the models were assigned non-zero weights, in absolute values and in percentage. The data were taken on 50 random prediction days for value υ set to 1 day. The model numbers correspond to Table 6.1.

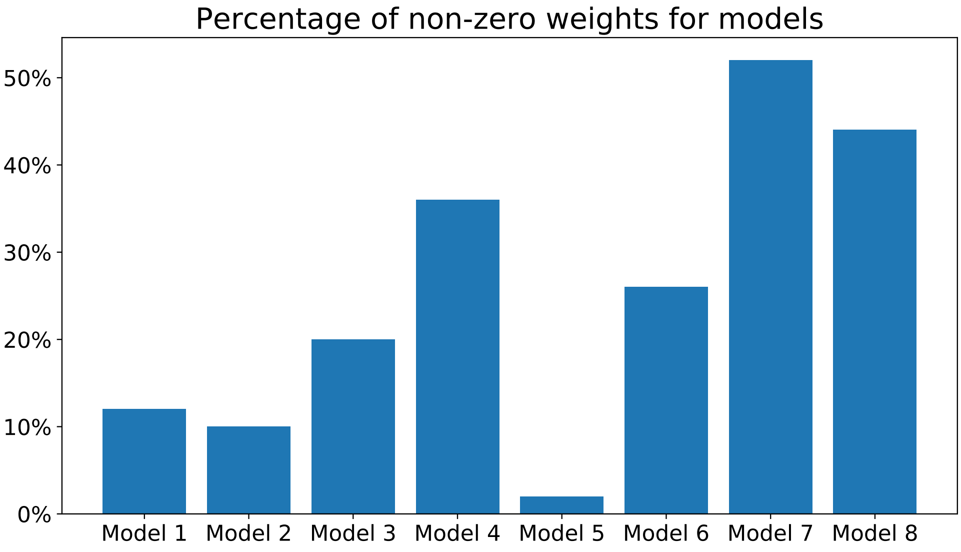


Figure 6.4: The percentage of non-zero weights for models. The values represent the percentage of times, the models were assigned non-zero weights. The data were taken on 50 random prediction days for value υ set to 1 day. The model numbers correspond to Table 6.1.

Conclusion

The goal of this thesis was to create predictive models for the electricity price forecasting (EPF), using weather conditions. This included the analysis of the available data for the weather and the electricity price, understanding the mechanisms of the electricity market, proposing convenient models, and evaluating their predictive performance.

In the theoretical part of this thesis, the necessary theoretical background was presented, and the existing techniques for the EPF were reviewed. Then, the available electricity prices and weather data were analyzed. Data from the EEX Group power market for Germany were selected for the modeling. Germany was determined to be a convenient country for predicting the electricity price based on weather conditions. The reason for it is that a large portion of German electricity is generated from renewable sources, which makes the electricity prices dependent on the weather. Weather forecasts from the GFS model were used, mainly because the data are available for free in the public domain.

The weather data were preprocessed and transformed into single time series for each element using a weighted average based on installed electrical capacity of solar and wind power plants in the nodes of $0.5°\times0.5°$ grid covering the whole of Germany.

For the modeling, statistical and machine learning models were chosen to be used. In particular, the selected models were the SARIMAX and the artificial neural networks. For improving the forecasts, an algorithm for averaging the predictions from multiple models was proposed, along with its probabilistic interpretation.

All of the proposed models and the averaging algorithm were implemented using `python3` programming language. The parameters of the models were tuned and a total of 8 models (6 SARIMAX, 2 ANN) were selected to be used in the model averaging algorithm. On a sample of 50 randomly selected days, the ensemble model achieved better average accuracy (in terms of RMSE) than any of the 8 individual models.

All of the goals of this thesis were fulfilled. In the future, the results could probably be improved by an exhaustive parameter tuning, which was not possible in this thesis due to the limited computational power. Also, more different types of models could be added to the averaging procedure, which would introduce more diversity into the ensemble model and possibly lead to better results. Additional exogenous regressors could also be added to the dataset, for example, coal prices (which affect the price through electricity generated in coal power plants) or the prices of emission allowances.

The results of this work are planned to be published in a scientific journal after further research.

Bibliography

[1] Rudnick, H. The Electric Market Restructuring in South America: Successes and Failures on Market Design. *Power Engineering Review, IEEE*, volume 18, July 1998.

[2] Domah, P.; Pollitt, M. G. The Restructuring and Privatisation of Electricity Distribution and Supply Businesses in England and Wales: A Social Cost-Benefit Analysis. *Fiscal Studies*, volume 22, no. 1, 2001: pp. 107–146.

[3] Sweeney, J. L. The California Electricity Crisis: Lessons for the Future. *National Academy of Engineering*, volume 32, no. 2, 2002: pp. 23–31.

[4] Zwillinger, D. *CRC Standard Probability and Statistics Tables and Formulae*. CRC Press, 1999.

[5] Akaike, H. Information theory and an extension of the maximum likelihood principle. In *2nd International Symposium on Information Theory*, 1973, pp. 267–281.

[6] Burnham, K. P. *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. Springer, 2003.

[7] Bishop, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 2011.

[8] Florescu, I. *Probability and Stochastic Processes*. Wiley, 2014.

[9] DenisBoigelot. An example of the correlation of x and y for various distributions of (x,y) pairs. 2011, [cit. 2019-04-29]. Available from: `https://en.wikipedia.org/wiki/File:Correlation_examples2.svg`

[10] Box, G. E. P.; Jenkins, G. M. *Time Series Analysis: Forecasting and Control, Fourth edition*. Wiley, 2008.

[11] Hyndman, R. J. *Forecasting: principles and practice*. OTexts, 2013.

[12] Statsmodels developers. SARIMAX documentation [online]. [cit. 2019-04-24]. Available from: `https://www.statsmodels.org/stable/generated/statsmodels.tsa.statespace.sarimax.SARIMAX.html`

[13] Haykin, S. O. *Neural Networks and Learning Machines (3rd Edition)*. Pearson, 2009, ISBN 0131471392.

[14] Weron, R. *Modeling and Forecasting Electricity Loads and Prices: A Statistical Approach*. Wiley, December 2006, ISBN 047005753X.

[15] Weron, R. Electricity price forecasting: A review of the state-of-the-art with a look into the future. *International Journal of Forecasting*, volume 30, no. 4, 2014: pp. 1030 – 1081.

[16] Willems, B.; Rumiantseva, I.; et al. Cournot versus Supply Functions: What does the data tell us? *Energy Economics*, volume 31, no. 1, 2009: pp. 38–47.

[17] Contreras, J.; Espinola, R.; et al. ARIMA Models to Predict Next-Day Electricity Prices. *Power Engineering Review, IEEE*, volume 22, October 2002.

[18] Weron, R.; Misiorek, A. Forecasting Spot Electricity Prices With Time Series Models. Technical report, University Library of Munich, Germany, 2005.

[19] Cuaresma, J. C.; Hlouskova, J.; et al. Forecasting electricity spot-prices using linear univariate time-series models. *Applied Energy*, volume 77, no. 1, 2004: pp. 87 – 106, ISSN 0306-2619.

[20] Huurman, C.; Ravazzolo, F.; et al. The power of weather. *Computational Statistics & Data Analysis*, volume 56, no. 11, 2012: pp. 3793 – 3807, 1st issue of the Annals of Computational and Financial Econometrics Sixth Special Issue on Computational Econometrics.

[21] Singhal, D.; Swarup, K. Electricity price forecasting using artificial neural networks. *International Journal of Electrical Power & Energy Systems*, volume 33, no. 3, 2011: pp. 550 – 555.

[22] Informationsportal Erneuerbare Energien. Time series for the development of renewable energy sources in Germany 1990 - 2018 [online]. [cit. 2019-04-13]. Available from: `https://www.erneuerbare-energien.de/EE/Navigation/DE/Service/Erneuerbare_Energien_in_Zahlen/Zeitreihen/zeitreihen.html`

[23] EEX Group. EEX Homepage [online]. [cit. 2019-04-13]. Available from: https://www.eex.com/

[24] EEX Group. EEX Group - Annual Report 2017. https://www.eex.com/dl/en/about/eex/annual-report/81436/file, April 2018.

[25] National Centers for Environmental Information (NOAA). Global Forecast System (GFS) [online]. [cit. 2019-04-13]. Available from: https://www.ncdc.noaa.gov/data-access/model-data/model-datasets/global-forcast-system-gfs

[26] Deutscher Wetterdienst. ICON (Icosahedral Nonhydrostatic) Model [online]. [cit. 2019-04-13]. Available from: https://www.dwd.de/EN/research/weatherforecasting/num_modelling/01_num_weather_prediction_modells/icon_description.html

[27] Recherche en Prevision Numerique. The Global Environmental Multiscale Model [online]. [cit. 2019-04-13]. Available from: http://collaboration.cmc.ec.gc.ca/science/rpn/gef_html_public/index.html

[28] Naval Oceanography Portal. The Fleet Numerical Meteorology and Oceanography Center (FNMOC) [online]. [cit. 2019-04-13]. Available from: https://www.usno.navy.mil/FNMOC

[29] Berger, E. The US weather model is now the fourth best in the world [online]. June 2016, [cit. 2019-04-13]. Available from: https://arstechnica.com/science/2016/06/the-us-weather-model-is-now-the-fourth-best-in-the-world/

[30] Raftery, A. E.; Madigan, D.; et al. Bayesian Model Averaging for Linear Regression Models. *Journal of the American Statistical Association*, volume 92, no. 437, March 1997: pp. 179–191.

[31] National Centers for Environmental Information (NOAA). Model Data [online]. [cit. 2019-04-16]. Available from: https://nomads.ncdc.noaa.gov/data/gfs4/

[32] Whitaker, J. Module pygrib. 2010. Available from: https://jswhit.github.io/pygrib/docs/

[33] Open Power System Data. Renewable power plants [online]. [cit. 2019-04-17]. Available from: https://data.open-power-system-data.org/renewable_power_plants/

[34] Pandas developers. pandas: Python Data Analysis Library. [cit. 2019-04-30]. Available from: https://pandas.pydata.org/

[35] NumPy developers. NumPy. [cit. 2019-04-30]. Available from: https://www.numpy.org/

[36] The Matplotlib development team. Matplotlib: Python plotting. [cit. 2019-04-30]. Available from: https://matplotlib.org/

[37] Hatalis, K. Tutorial: Multistep Forecasting with Seasonal ARIMA in Python. [cit. 2019-04-30]. Available from: https://www.datasciencecentral.com/profiles/blogs/tutorial-forecasting-with-seasonal-arima

[38] scikit-learn developers. MLPRegressor. [cit. 2019-04-28]. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html

[39] The SciPy community. scipy.optimize.minimize. [cit. 2019-05-01]. Available from: https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html

Appendix A

Acronyms

ACF Autocorrelation function

AIC Akaike information criterion

ANN Artificial neural networks

ARIMA Autoregressive integrated moving average

ARMA Autoregressive moving average

C(E)ST Central European (Summer) Time

EEX European Energy Exchange

EPF Electricity price forecasting

MCP Market clearing price

ML Machine learning

NWP Numerical weather prediction

PACF Partial autocorrelation function

PCC Pearson correlation coefficient

RNN Recurrent neural networks

SARIMA Seasonal autoregressive integrated moving average

SARIMAX Seasonal autoregressive integrated moving average with eXogenous regressors

SVM Support vector machines

UTC Coordinated Universal Time

Appendix **B**

Contents of enclosed CD

```
readme.txt ........................ the file with CD contents description
data ................................. the directory with code and data
    capacities .................. the data about power plant capacities
    data_downloading ................. the scripts for data downloading
    prices ................................. the electricity price data
    weather ....................................... the weather data
doc ................... the directory of LaTeX source codes of the thesis
    images ............................... the directory with images
    MT_Dejdar_Jan_2019.pdf .............. the thesis text in PDF format
```