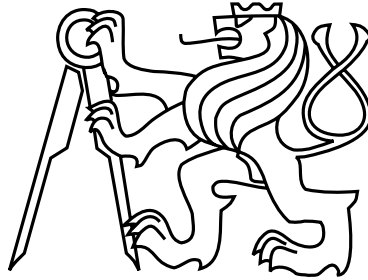


Czech Technical University in Prague  
Faculty of Electrical Engineering  
Department of Computer Science



Master's Thesis

## **Time Series Data Prediction and Analysis**

*Oleg Ostashchuk*

Supervisor: Ing. Miloš Kozák, Ph.D.

Study Programme: Open Informatics

Field of Study: Software Engineering

May 2017

Czech Technical University in Prague  
Faculty of Electrical Engineering

Department of Computer Science

## DIPLOMA THESIS ASSIGNMENT

Student: **Bc. Oleg Ostashchuk**

Study programme: Open Informatics  
Specialisation: Software Engineering

Title of Diploma Thesis: **Prediction Time Series Data Analysis**

### Guidelines:

1. Survey existing methods of time series analysis for prediction of time series behavior.
2. Study provided data, and select at least 3 methods for analysis. Discuss necessary data pre-processing for each suggested method.
3. Implement selected methods for data pre-processing and data analysis for each data set.
4. Extend selected dataset and method in order to improve accuracy of prediction for specified prediction period.
5. Evaluate the accuracy of implemented method and discuss an improvement.

### Bibliography/Sources:

- [1] HILLIER, F.S.; LIEBERMAN, G.J.: Introduction to operations research. 7th Edition. Boston: McGraw-Hill, 2001, 1214 p. ISBN: 0-07232-169-5.
- [2] HAYKIN, S.O.: Neural networks and learning machines. 3rd Edition. New York: Prentice Hall, 2009, 936 p. ISBN: 0-13147-139-2.
- [3] MCKINNEY, W.: Python for data analysis. Beijing: O'Reilly Media, 2013, 466 p. ISBN: 1-44931-979-3.

Diploma Thesis Supervisor: Ing. Miloš Kozák, Ph.D.

Valid until the end of the summer semester of academic year 2016/2017



Prague, February 17, 2016

# Acknowledgements

I would like to thank my supervisor Ing. Miloš Kozák, Ph.D. for his help, spent time, as well as his kindly approach for several years of collaboration during bachelor's and master's degree studies.



## Declaration

I hereby declare that I have completed this thesis independently and that I have listed all the literature and publications used.

I have no objection to usage of this work in compliance with the act §60 Zkon . 121/2000Sb. (copyright law), and with the rights connected with the copyright act including the changes in the act.

In Prague on May 26, 2017

.....



# Abstract

Given thesis deals with the problematic of time series analysis and forecasting. The aim of thesis is to survey an existing time series forecasting methods, including necessary data preprocessing steps. There are selected three promising forecasting methods, including ARIMA method, artificial neural networks method and double exponential smoothing method.

The main task of the thesis, is to perform data analysis of provided data and to develop the individual forecasting models.

At the end of the thesis, there are results summary and further improvements are discussed.

# Abstrakt

Diplomová práce se věnuje problematice analýzy a prognózování časových řad. Cílem práce je prozkoumat existující metody prognózování časových řad, včetně potřebných kroků předzpracování dat. Jsou vybrány tři slibné metody prognózování, včetně ARIMA, metody prognózování pomocí Neuronových sítí a metody dvojitého exponenciálního vyrovnání.

Dále je v práci provedena analýza nabodnutých dat a jsou zkonstruovaný jednotlivé modely prognózování.

V závěru práce je provedené zhodnocení výsledku a jsou uvedené perspektivy pro další vylepšení kvality predikce.

# Contents

<b>List of Figures</b>	<b>x</b>
<b>List of Tables</b>	<b>xii</b>
<b>Abbreviations</b>	<b>xiii</b>
<b>I Theoretical part</b>	<b>1</b>
<b>1 Introduction</b>	<b>2</b>
1.1 Aims of the Thesis . . . . .	2
<b>2 Time Series Analysis</b>	<b>4</b>
2.1 Introduction to Time Series . . . . .	4
2.2 Time Series Types Classification . . . . .	5
2.3 Aims of Time Series Analysis . . . . .	6
2.4 Time series components . . . . .	7
2.5 Autocorrelation and Partial Autocorrelation . . . . .	9
2.5.1 Autocorrelation function . . . . .	9
2.5.2 Partial Autocorrelation function . . . . .	9
2.6 Time series forecasting . . . . .	10
2.6.1 Forecasting without external factors . . . . .	11
2.6.2 Forecasting with external factors . . . . .	13
2.7 Forecasting Accuracy . . . . .	14
2.8 Data preprocessing . . . . .	15
2.8.1 Outliers detection . . . . .	15
2.8.2 Denoising and Smoothing . . . . .	16
2.8.3 Differencing . . . . .	17
2.8.4 Scaling . . . . .	17
2.8.5 Normalization . . . . .	18
<b>3 Forecasting Methods</b>	<b>19</b>
3.1 Regression models . . . . .	19
3.2 Autoregressive and moving average models . . . . .	20
3.3 Exponential smoothing models . . . . .	21
3.3.1 Double exponential smoothing . . . . .	22
3.4 Artificial neural networks models . . . . .	23
3.4.1 Biological inspiration . . . . .	23



---

3.4.2	Artificial neuron model . . . . .	24
3.4.3	Types of Activation Function . . . . .	26
3.4.4	Neural Network Architectures . . . . .	27
3.4.5	Appropriate architecture . . . . .	28
3.4.6	Networks training . . . . .	29
3.4.7	Cross-validation . . . . .	29
3.4.8	ANN forecasting model . . . . .	31
3.5	Markov chain models . . . . .	31
3.6	Forecasting models comparison . . . . .	32
<b>II</b>	<b>Practical part</b>	<b>34</b>
<b>4</b>	<b>Internet traffic data forecasting</b>	<b>35</b>
4.1	Internet traffic data set experiments . . . . .	35
4.2	Data analysis and preprocessing . . . . .	36
4.3	Autoregressive-Moving-average method . . . . .	39
4.4	Artificial neural networks method . . . . .	41
4.5	Exponential smoothing method . . . . .	43
4.6	Experiments summary . . . . .	44
<b>5</b>	<b>Main experiments</b>	<b>45</b>
5.1	Data analysis and preprocessing . . . . .	46
5.2	Autoregressive-Moving-average method . . . . .	48
5.2.1	Forecasting with external factors . . . . .	50
5.3	Exponential smoothing method . . . . .	51
5.4	Artificial neural networks method . . . . .	52
5.4.1	Forecasting without external factors . . . . .	53
5.5	Forecasting with external factors . . . . .	54
5.6	Data set extension . . . . .	56
5.7	Experiments summary . . . . .	57
<b>6</b>	<b>Conclusion</b>	<b>61</b>
6.1	Discussion about further improvements . . . . .	62
	<b>Bibliography</b>	<b>63</b>

# List of Figures

2.1	Seismogram from HAWA station (Source: Hanford, Washington, USA) . . .	5
2.2	Monthly international airline passenger counts from 1949 to 1960 . . . . .	7
2.3	Time series components . . . . .	8
2.4	ACF and PACF of monthly airline passenger counts . . . . .	10
2.5	Time series forecasting without external factors . . . . .	12
2.6	Time series forecasting with external factors . . . . .	14
2.7	Outliers detection example . . . . .	16
3.1	Biological neuron . . . . .	24
3.2	Artificial neuron model . . . . .	25
3.3	Activation functions . . . . .	26
3.4	Artificial neural network example . . . . .	27
3.5	Artificial neural network example . . . . .	28
3.6	Overfitting example . . . . .	30
3.7	Cross-validation . . . . .	30
3.8	ANN and time series forecasting . . . . .	31
3.9	Markov chain model . . . . .	32
4.1	Plot of the internet traffic data (in bits) . . . . .	36
4.2	Plot of the internet traffic data (in bits) . . . . .	37
4.3	The plot of ACF of raw time series data. . . . .	37
4.4	The ACF plot of internet traffic data after log transformation, one non-seasonal differencing and one non-seasonal differencing with lag=24. . . . .	38
4.5	The PACF plot of internet traffic data after log transformation, one non-seasonal differencing and one non-seasonal differencing with lag=24. . . . .	39
4.6	The plots of one hour ahead forecasted values (blue) and actual values (red) of the test data set. . . . .	40
4.7	Diagram of ANN used in this experiment. It demonstrates structure of network, number of neurons, activation functions and dropout regularization parameters of individual layers. . . . .	42
4.8	The plots of one hour ahead forecasted values (blue) and actual values (red) of the test data set. . . . .	42
4.9	The plots of one hour ahead forecasted values (blue) and actual values (red) of the test data set. . . . .	43
5.1	Plot of the local outdoor temperature data (in degree Celsius) . . . . .	46
5.2	ACF plot of the local outdoor temperature data . . . . .	47
5.3	ACF plot of the local outdoor temperature data after one non-seasonal differencing . . . . .	47

---

5.4	The plot of preprocessed local outdoor temperature data . . . . .	48
5.5	ACF plot of preprocessed local outdoor temperature data . . . . .	49
5.6	PACF plot of preprocessed local outdoor temperature data . . . . .	49
5.7	The plots of one hour ahead forecasted values (blue) and actual values (red) of the test data set. . . . .	51
5.8	The plots of one hour ahead forecasted values (blue) and actual values (red) of the test data set. . . . .	52
5.9	Diagram of ANN used in this experiment. It demonstrates structure of network, number of neurons, activation functions and dropout regularization parameters of individual layers. . . . .	54
5.10	Diagram of ANN used in this experiment. It demonstrates structure of network, number of neurons, activation functions and dropout regularization parameters of individual layers. . . . .	55
5.11	The plots of one hour ahead forecasted values (blue) and actual values (red) of the test data set. . . . .	56
5.12	Example of artificially generated time series data . . . . .	57
5.13	The plot of forecasting errors calculated on the test data set by the ANN forecasting model with the external factor. . . . .	59
5.14	The plot of forecasting errors calculated on the test data set by the ANN forecasting model with the external factor. . . . .	60

# List of Tables

4.1	Forecasting performance of $SARIMA(9, 1, 8) \times (0, 1, 1)$ model for different forecast horizons. . . . .	40
4.2	Forecasting performance of ANN model for different forecast horizons. . .	42
4.3	Forecasting performance of Double exponential smoothing model for different forecast horizons. . . . .	43
5.1	Forecasting performance of $SARIMA(9, 1, 5) \times (0, 1, 1)$ model for different forecast horizons. . . . .	50
5.2	Forecasting performance of $SARIMAX(9, 1, 5) \times (0, 1, 1)$ model for different forecast horizons. . . . .	51
5.3	Forecasting performance of double exponential smoothing model for different forecast horizons. . . . .	52
5.4	Forecasting performance of ANN model for different forecast horizons. . .	54
5.5	Forecasting performance of ANN model for different forecast horizons, after adding an external factor . . . . .	56

# Abbreviations

<b>ACF</b>	<b>A</b> utocorrelation <b>F</b> unction
<b>PACF</b>	<b>P</b> artial <b>A</b> utocorrelation <b>F</b> unction
<b>NN</b>	<b>N</b> eural <b>N</b> etwork
<b>ANN</b>	<b>A</b> rtificial <b>N</b> eural <b>N</b> etwork
<b>FFNN</b>	<b>F</b> eed <b>F</b> orward <b>N</b> eural <b>N</b> etwork
<b>RNN</b>	<b>R</b> ecurrent <b>N</b> eural <b>N</b> etwork
<b>LSTM</b>	<b>L</b> ong <b>s</b> hort-term memory
<b>DES</b>	<b>D</b> ouble <b>E</b> xponential <b>S</b> smoothing
<b>LP</b>	<b>L</b> inear <b>P</b> rogramming
<b>MAPE</b>	<b>M</b> ean <b>A</b> bsolute <b>P</b> ercentage <b>E</b> rror
<b>RMSE</b>	<b>R</b> oot <b>M</b> ean <b>S</b> quared <b>E</b> rror
<b>MSE</b>	<b>M</b> ean <b>S</b> quared <b>E</b> rror



## Part I

# Theoretical part

# Chapter 1

## Introduction

The word "prediction" originates from a Latin statement "praedicere", which was originally denoted by meanings "to say beforehand" or "to mention in advance". Today, "prediction" is usually referred to some kind of message or opinion about an event that is expected to happen in future. Inside the more formal science context, the process of making predictions about future by using scientific methods is usually denoted by term "forecasting". Processes that are usually required to be forecasted, are the most often stored in a so called time series format. [1]

Time series is a common mathematical expression that can be frequently observed in various texts about statistics, signal processing or econometrics. Every day, newspapers contain business sections, which report daily stock prices, foreign currency exchange rates or monthly rates of unemployment. Meteorology records usually consists of hourly wind speeds, daily maximum and minimum temperatures or annual rainfall. Geophysics are continuously observing processes like shaking or trembling of the earth, in order to predict possibly impending earthquakes. All these and certainly many other examples could be mentioned to describe the role of time series in our society. [2]

### 1.1 Aims of the Thesis

Today, there is plenty of various forecasting methods and each of them requires the corresponding conditions and proper data preprocessing. Performing a research in the given problematic, it can be observed, that the autoregressive methods and exponential smoothing belong to the most frequently used forecasting methods. Additionally, an Artificial intelligence, especially artificial neural networks demonstrate a great success with the assigned tasks, including the time series forecasting.



The main issue of this thesis is to perform the analysis of provided data and to develop the qualitative forecasting models for them. In order to solve this task, the theoretical part of the thesis will be devoted to the survey of the time series problematic, forecasting methods, data preprocessing and other important aspects of time series analysis.

## Chapter 2

# Time Series Analysis

### 2.1 Introduction to Time Series

The term "time series" itself, denotes a data storing format, which consists of the two mandatory components - time units and the corresponding value assigned for the given time unit. Values of the series need to denote the same meaning and correlate among the nearby values. Restriction is, that at the same time there can be at most one value for each time unit. For example, sequences, which just enumerate some values, they do not fulfill the time series requirements.

In theory, there are two fundamental ways, how time series data are recorded. The first way, values are measured just for the specific timestamps, what may occur periodically, or occasionally according to concrete conditions, but anyway, result will be a discrete set of values, formally called discrete time series. This is very common case and frequently observed in practice. In economy sector, most of the indicators are measured periodically with the specific periods, therefore economic indicators represent an appropriate example of discrete time series.

The second option is, that data are measured and recorded continuously along the time intervals. Electrical signals from sensors, earth shakings, various indicators from medicine, like ECG, or many other scientific sensors, they all represent a continuous measurement of corresponding physical quantity. This kind of processes produces a continuous time series. Figure 2.1 demonstrates a seismogram from station HAWA (Hanford, Washington, USA), example of continuous time series.

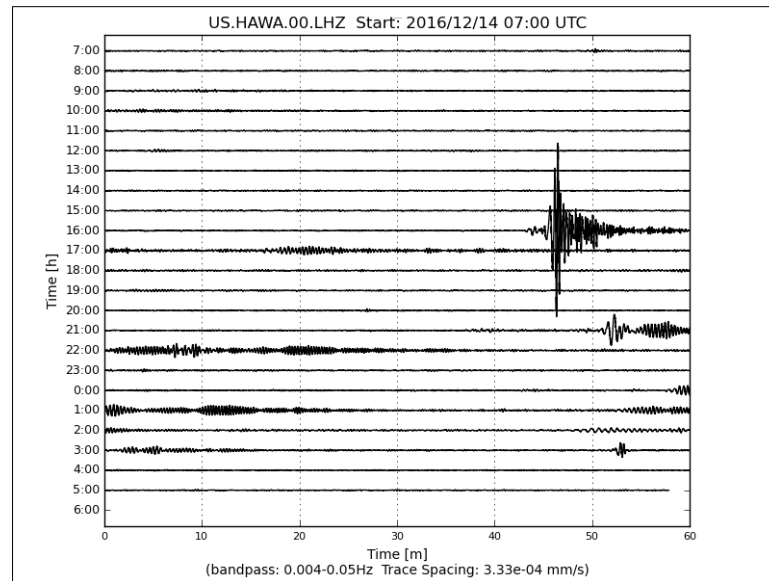


FIGURE 2.1: Seismogram from HAWA station (Source: Hanford, Washington, USA)

## 2.2 Time Series Types Classification

There are many various time series classifications based on specific criteria. The most significant dependencies are: length of the time step, memory and stationarity.

Depending on the distance between recorded values, time series data are classified into:

- equidistant time series
- non-equidistant time series

Equidistant time series are formed, when its values are recorded periodically with a constant period length. A lot of physical or environmental processes are described by this kind of time series. Non-equidistant time series are those time series, which do not keep the constant distance between observations. Econometric indicators, like stock prices are not necessary performed within regular time intervals, they are regulated by a concrete supply and demand rates on the specific market. Therefore, this kind of series suitably demonstrates a non-equidistant time series example.

According to the rate of dependency between newly observed values and its predecessors, time series are divided into:

- long memory time series
- short memory time series

Time series with long memory are those, for which the autocorrelation function decreases slowly. [1] This kind of time series usually describes processes, which don't have fast turnovers. Traffic congestion, electric energy consumption, different physical or meteorological indicators, like air temperature measurements, all these processes are usually described by long memory time series. Short memory time series are those, for which autocorrelation function is decreasing more rapidly. Typical examples contain processes from the econometric sector.

Another classification of time series is based on their stationarity:

- stationary time series
- non-stationary time series

Stationary time series are time series, for which statistical properties like mean value or variance, are constant over time. These time series stay in relative equilibrium in relation to its corresponding mean values. Other time series belong to non-stationary time series. In industry, trading or economy, time series more frequently belongs to the non-stationary category. In order to deal with the forecasting task, non-stationary time series are usually transformed to the stationary ones, by the appropriate preprocessing methods.

## 2.3 Aims of Time Series Analysis

Time series analysis unites a group of methods for work with time series data, in order to extract the potentially useful information. There are two main goals of time series analysis:

1. Determination of the time series behavior - Identification of the important parameters and characteristics, which adequately describe the time series behavior.
2. Time series forecasting - Forecasting the future values of the time series, depending on its actual and past values.

Both of these goals require the time series model identification. As soon as the model is indentified, it can be exploited to interpret the time series behavior, for example, to understand the seasonal changes of the commodity prices. The model can also be used to extrapolate the time series, i.e. to forecast its future values.

## 2.4 Time series components

Usually, the most of analysis methods assume, that time series data contains the systematic component (typically comprising several components) and random noise (error), which complicates detection of the regular components. Therefore, the majority of methods, includes different noise filtration methods, in order to detect the regular components, or it has to performed during data preprocessing.

The most of the regular components belongs to two main classes. They belong to either a trend or seasonal component. The trend is a general systematic linear or non-linear component, which may change over time. Seasonal component is periodically repeating component. Both these types of regular components are usually presented in the time series simultaneously. For example, sales may increase from year to year, but there is a seasonal component, which reflects the significant growth of sales in December and a drop in August.

This model can be demonstrated on the series representing the monthly international airline passenger counts from 1949 to 1960. The graph of monthly passenger counts clearly demonstrates almost linear trend, i.e. stable increase from year to year (the number of transported passengers in 1960 is four times greater, than in 1949). In the same time, the progress of monthly rates within one year is repeating, and is similar from year to year (for example, the rate of passengers is higher in the periods of holidays).

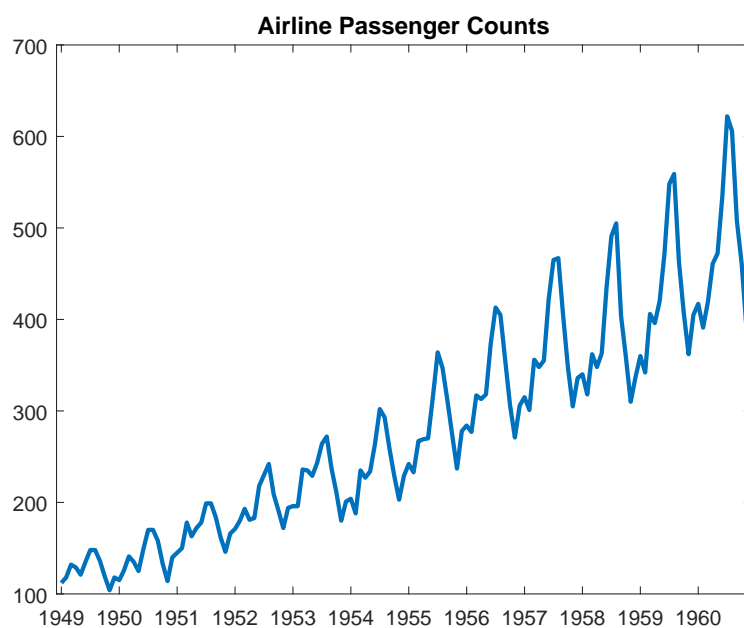


FIGURE 2.2: Monthly international airline passenger counts from 1949 to 1960

It has been already mentioned, that general model of time series usually contains several components: trend component  $T(t)$ , seasonal component  $S(t)$ , random noise component  $R(t)$ , and sometimes there is additionally mentioned a cyclical component  $C(t)$ . The difference between cyclical and seasonal components is, that seasonal components represents a regular seasonal periodicity, while cyclical component has a longer lasting effect and may vary from cycle to cycle. Very often, cyclical component is integrated into one trend component  $T(t)$ . Figure 2.3 demonstrate an example of time series decomposition.

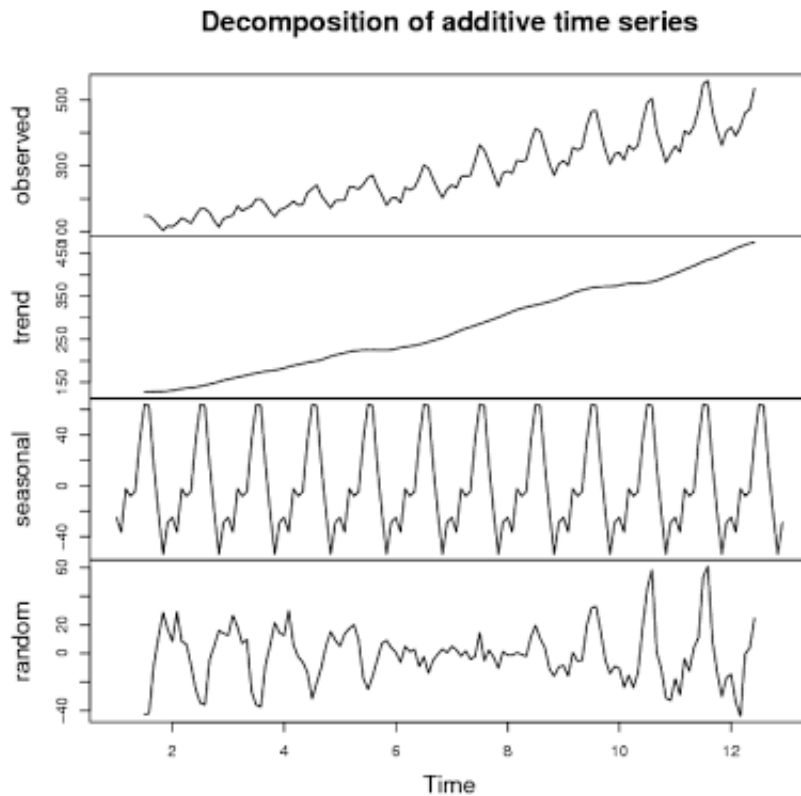


FIGURE 2.3: Time series components

Now, it is important to describe, how this components mathematically interact together, in order to compose a time series. The concrete functional relationships between the components may vary for different series. However, there are two main models, how they interact to each other:

- Additive model

$$Z(t) = T(t) + C(t) + S(t) + R(t) \quad (2.1)$$

- Multiplicative model

$$Z(t) = T(t) \times C(t) \times S(t) \times R(t) \quad (2.2)$$

Main difference between these two models may be observed in a growth rate. Previously mentioned example of monthly airline passenger counts, demonstrates a typical multiplicative model, where the amplitude of seasonal changes increases with the trend. The growths of the trend or seasonal components may be expressed in percentage (multiplicative model) or in absolute values (additive model). [2]

## 2.5 Autocorrelation and Partial Autocorrelation

Dependencies between the actual and historical values represent a fundamental principle of time series forecasting. It can be easily observed, that each value of the series is very similar to its neighboring values. Additionally, time series contain a seasonal component, what means, that each value is also dependent on the values of identical time, but one season ago. Formally, any statistical dependency between two entities is denoted as a correlation, and is expressed by a corresponding coefficient.

### 2.5.1 Autocorrelation function

Autocorrelation function calculates the correlations between the time series and its shifted copies at different points in time. The autocorrelations are usually calculated for the specific range of lags (shifts) and are expressed in the form of graph, called correlogram. Investigation of autocorrelations, enables to detect important dependencies in time series data. [3]

### 2.5.2 Partial Autocorrelation function

Sometimes it can happen, that the first value is heavily dependent on the second value, the second value is heavily dependent on the third value and therefore the first value is also dependent on the third, and so on. This causes, that significant dependencies can be not found on the graph of autocorrelation function. Partial autocorrelation function is another important tool. It is a modification of autocorrelation function, which allows to eliminate the described problem. [4]

Figure 2.4 demonstrates results of autocorrelation function and partial autocorrelation function for the time series data from the previous section (Monthly international airline passenger counts from 1949 to 1960).

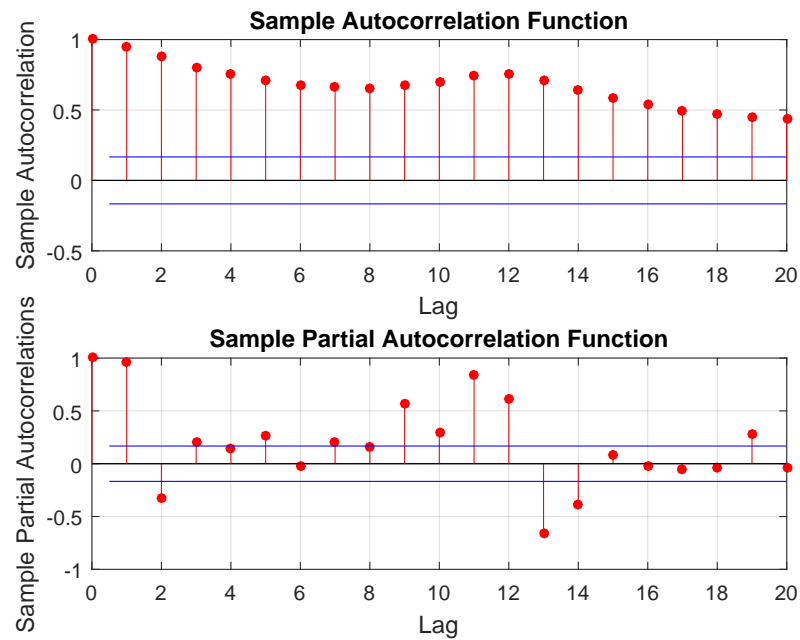


FIGURE 2.4: ACF and PACF of monthly airline passenger counts

## 2.6 Time series forecasting

Time series forecasting belongs to most important analysis methods, performed over the time series data. General idea is based on the fact, that information about the past events can be effectively exploited to create predictions about the future events. From the point of view of the time series data, this means, that forecasting models use already measured values to predict future values before they are observed.

When talking about the time series forecasting, it is necessary to emphasize the importance of distinction between two terms, "forecasting methods" and "forecasting models". Despite the fact, that both these terms have precisely specified meaning, in practice, they are often used mistakenly with the mixed meanings.

- Forecasting method – Denotes an algorithmic sequence of actions, that are necessary to perform, in order to obtain the time series forecasting model. Additionally, forecasting methods determines the way of quality assessment measurements.
- Forecasting model – Denotes a functional representation, that adequately describes a time series. On the basis of this forecasting model, future values of the time series are forecasted.



There are two main ways, how the time series forecasting tasks are defined. The first option is based on the computations, that use only the past values of the same time series, in order to predict the values in future. The second option allows to use not only the past values of the same time series, but also another external factors in addition, that can be useful for forecasting. In these cases, external factors are very often presented as another time series. Time series of the external factors are not obliged to have the same time step intervals, as the original time series data. Therefore, additional steps must be taken, in order to deal with this problem. It is also expected, that the external factors should have some influence on the original time series progress. For example, an intuitive external factors of energy consumption could be various meteorological indicators, like air temperature or air humidity.

### 2.6.1 Forecasting without external factors

Time series forecasting without external factors. If the observations of some stochastic process are available at discrete units of time  $t = \{1, 2, \dots, T\}$ , then the sequence of values  $Z(t) = \{Z(i) \mid i \in T\} = \{Z(1), Z(2), \dots, Z(T)\}$  is denoted as a time series.

Let's assume that at the moment of time unit  $-T$ , it is necessary to make a forecast of  $-l$  future values of the given process  $Z(t)$ . In other words, it is needed to determine the most probable future values for each of the time units  $\{T+1, \dots, T+l\}$ . Time unit  $-T$  is a moment when the forecast is performed, it is usually named by term "**origin**". The parameter  $-l$  is denoted as a "**leadtime**", it represents the number of future values that are going to be predicted.

In order to calculate the time series values at future time units, it is necessary to determine functional dependency that describes a relationship between past and future values of the given time series. The forecast is based on  $-k$  past values, denoted as an input vector  $Z_T$ . As a result, the vector of  $-l$  future predictions will be obtained, denoted as an output vector  $\hat{Z}_T$ . All predicted values  $\hat{Z}(i)$  will be marked with sign  $\hat{\phantom{z}}$  in order to label them as predictions, not the real values.

$$Z_T = \begin{pmatrix} Z(T) \\ Z(T-1) \\ Z(T-2) \\ \vdots \\ Z(T-k) \end{pmatrix} \quad \hat{Z}_T = \begin{pmatrix} \hat{Z}(T+1) \\ \hat{Z}(T+2) \\ \vdots \\ \hat{Z}(T+l) \end{pmatrix} \quad (2.3)$$

$$f(Z_T) = \hat{Z}_T \quad (2.4)$$

The functional dependency (2.2) is usually denoted as forecast function and it represents the forecast model. The intuitive aim is to find the forecast function such that the deviations between predicted values and actual values, that will be observed later in future, are as small as possible.

$$\varepsilon_T = \begin{pmatrix} Z(T+1) \\ Z(T+2) \\ \vdots \\ Z(T+l) \end{pmatrix} - \begin{pmatrix} \hat{Z}(T+1) \\ \hat{Z}(T+2) \\ \vdots \\ \hat{Z}(T+l) \end{pmatrix} \quad (2.5)$$

Analysis of deviations vector (2.3) represents a basis of so called “loss function” or “error function”. This function measures the quality of forecast, based on the measured deviations. There are more options, how to calculate rate of quality from the deviations vector, usually root mean square error or mean absolute deviation are calculated. More details about error functions will be discussed in section 2.2. The formal objective of time series forecasting is then formulated as a minimization of loss function.

In addition to calculations of future values, sometimes it is required to determine accuracy limits. The accuracy of the forecasts may be expressed by calculating probability limits on either side of each forecast. These limits may be calculated for any convenient set of probabilities. They are such that the realized value of the time series, when it eventually occurs, will be included within these limits with the stated probability. [1]

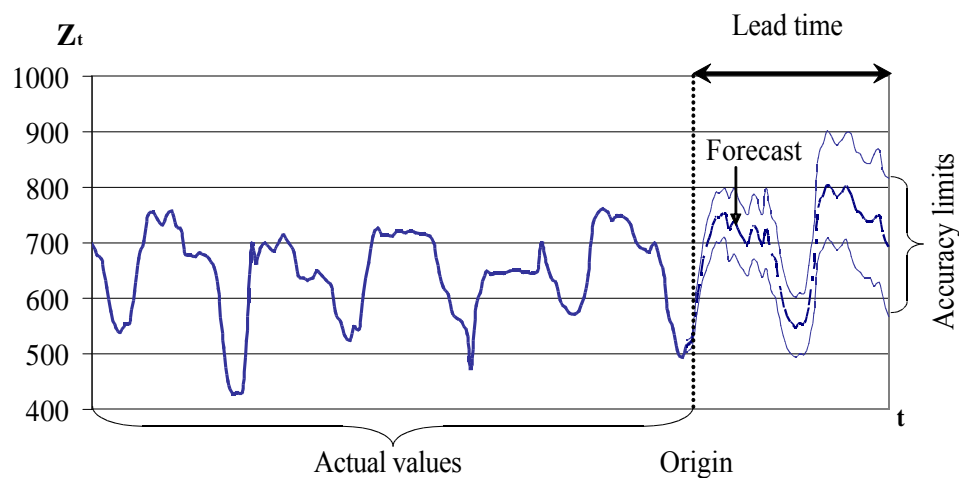


FIGURE 2.5: Time series forecasting without external factors

### 2.6.2 Forecasting with external factors

Time series process  $Z(t)$  is specified at the discrete time units  $t = \{1, 2, \dots, T\}$ . It is assumed, that this time series is affected by a set of external factors  $\{X_1(t_1), X_2(t_2), \dots, X_m(t_m)\}$ . Each external factor is represented as an independent time series process. For example, an external factor  $X_1(t_1)$  is specified at the corresponding discrete time units  $t_1 = \{1, 2, \dots, T_1\}$ .

The original time series  $Z(t)$  and external factors  $X_i(t_i)$  are not obliged to be specified at same time units. If the time units  $t, t_1, t_2, \dots, t_m$  are not equal, then it is necessary to recalculate the values of external factor to a single scale  $t$ .

Let's assume that at the moment of time unit  $T$ , it is necessary to make a forecast of  $-l$  future values of the given process  $Z(t)$ . In order to calculate the predictions, it is necessary to determine functional dependency, that describes a relationship between past and future values, also considering the impact of external factors.

$$Z_T = \begin{pmatrix} Z(T) \\ Z(T-1) \\ Z(T-2) \\ \vdots \\ Z(T-k) \end{pmatrix} \quad X_{i,T} = \begin{pmatrix} X_i(T+l) \\ \vdots \\ X_i(T+1) \\ X_i(T) \\ X_i(T-1) \\ \vdots \\ X_i(T-k) \end{pmatrix} \quad \hat{Z}_T = \begin{pmatrix} \hat{Z}(T+l) \\ \vdots \\ \hat{Z}(T+2) \\ \hat{Z}(T+1) \end{pmatrix} \quad (2.6)$$

$$f(Z_T, X_{1,T}, X_{2,T}, \dots, X_{m,T}) = \hat{Z}_T \quad (2.7)$$

The functional dependency (2.5) is a forecast function and it represents the forecast model with external factors. The rest tasks are performed in the same way as they were in the case of forecasting without external factors. The main objective is to find the forecast function such that the deviations between predicted values and actual values, that will be observed later in future, are as small as possible. This objective formulates minimization task of so called "loss function" or "error function". More details about error functions will be discussed in section 2.2.

The accuracy limits may be calculated for any convenient set of probabilities. Accuracy limits are such that the realized value of the time series, when it eventually occurs, will be included within these limits with the stated probability. [1]

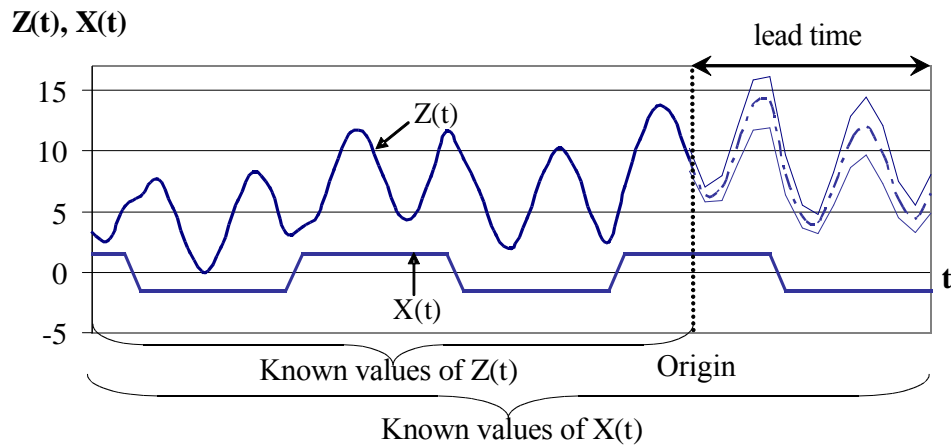


FIGURE 2.6: Time series forecasting with external factors

## 2.7 Forecasting Accuracy

Forecasting accuracy is a measure, which expresses performance of forecasting model. It is a reverse value to the measure of forecasting error. There are more options, how to calculate the measure of forecasting error. Each of them expresses a little bit different information. At the beginning, it is necessary to define the forecast error. It is expressed as a deviation of predicted value and actual value:

$$\varepsilon(t) = Z(t) - \hat{Z}(t) \quad (2.8)$$

- Mean absolute percentage error (MAPE)

$$MAPE = \frac{1}{N} \sum_{t=1}^N \frac{|Z(t) - \hat{Z}(t)|}{Z(t)} \cdot 100\% \quad (2.9)$$

- Root Mean squared error (RMSE)

$$RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^N (Z(t) - \hat{Z}(t))^2} \quad (2.10)$$

- Mean squared error (MSE)

$$MSE = \frac{1}{N} \sum_{t=1}^N (Z(t) - \hat{Z}(t))^2 \quad (2.11)$$

- Mean absolute error (MAE)

$$MAE = \frac{1}{N} \sum_{t=1}^N |Z(t) - \hat{Z}(t)| \quad (2.12)$$

- Sum of squared errors (SSE)

$$SSE = \sum_{t=1}^N (Z(t) - \hat{Z}(t))^2 \quad (2.13)$$

The suitability of MSE, RMSE, MAE and SSE measures is quite similar. They differ only a little bit, for example strong errors are penalized by RMSE less than by other measures. MAE and RMSE represent a scale dependent measure, while others are not scale dependent. All these measures are suitable for comparison of different forecasting methods on the same test data.

MAPE is one of the most frequently used forecasting error measures. It expresses the percentage error, what makes it easily understandable. It is a suitable measure for comparing the performance of one forecasting method on different testing data. But it has one significant shortcoming, it can be used only for time series with values much greater than 1. Otherwise, if the actual value of the series is close to 0, then a denominator will contain a very small number, what will make MAPE measure close to infinity. This will not express a correct performance.

## 2.8 Data preprocessing

Before the raw time series data can be applied to the forecasting methods, usually they have to undergo several transformations. Proper data preprocessing significantly affects the forecast quality. Some forecasting methods, for example neural networks methods, have strict requirements for the format of input data. The absence of the proper data preprocessing, leads to the inefficiency of the given forecasting method.

### 2.8.1 Outliers detection

An outlier is an observation, that significantly differs from the other observations in the sample. In practice, very often can be observed situation, when data contain some outliers. Identification of potential outliers is a very important preprocessing task, because of the following reasons:

1. Outlier may indicate mistakenly recorded data.
2. Sometimes the outlier may represent the correct data, but their presence decreases the effectiveness of the forecasting model. Therefore, their presence is undesired.

Outliers detection is usually performed by application of some appropriate filtering methods, for example "Hampel filter". [5] As soon as the outlier is detected, it can be excluded from the dataset, or replaced by the mean of its neighboring values.

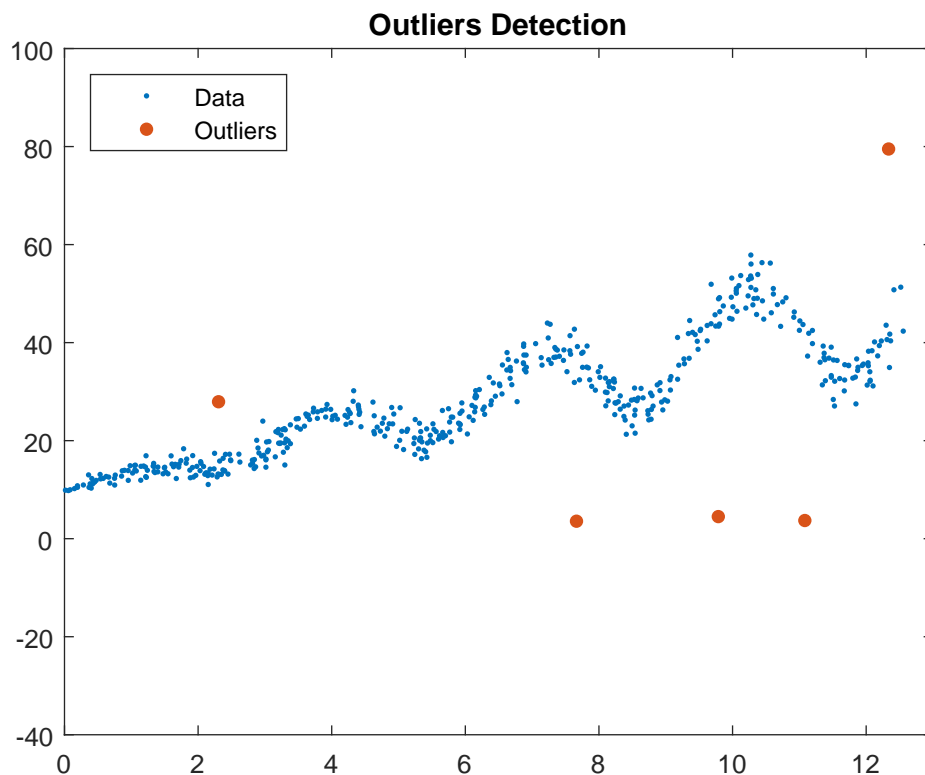


FIGURE 2.7: Outliers detection example

### 2.8.2 Denoising and Smoothing

Time series data almost always contain a random noise component. The purpose of denoising methods, is to filter and remove the unwanted noise. Smoothing of the processed data belongs to the most common denoising methods. Smoothing performs some kind of local averaging, which usually causes the elimination of unwanted noise signal. This can be explained by the fact, that random noise is known to be a stationary process, and stationary processes have a mean value equal to zero. Therefore, smoothing can be suitably used to remove the noise. The most popular smoothing algorithms are:

- Moving average filter - Each value in the series is replaced by the simple or weighted average of its neighboring values.
- Median filter - Similar to moving average, but values are replaced by median value.
- Local regression filter - Values are replaced by the smoothed curve with values fitted by least squares approach.

### 2.8.3 Differencing

In practice, very often happens, that it is necessary to forecast a non-stationary time series data. But the majority of forecasting methods can work only with the stationary series. There are several options, how this problem can be solved.

The most common option is differencing of the time series, which usually reduces the non-stationarity. Differencing can be performed multiple times, if there still remains some evidences of non-stationarity. Similarly the rate of relative differences can be used.

- Simple differencing:

$$D(t) = Z(t) - Z(t - 1) \quad (2.14)$$

- Relative differencing:

$$R(t) = \frac{Z(t) - Z(t - 1)}{Z(t - 1)} \quad (2.15)$$

Another option is application of logarithmic return rate. This is very similar method, it just use the logarithmic values instead of absolute values. Logarithmic return rate provides better scaling properties, which are useful if the original data contain an increasing oscillation character or exponential trend.

$$LR(t) = \log(Z(t)) - \log(Z(t - 1)) = \log\left(\frac{Z(t)}{Z(t - 1)}\right) \quad (2.16)$$

### 2.8.4 Scaling

Scaling is a transformation, that adjust scales of the values within some specific boundaries. The most common used scaling are transformations of values within  $\langle -1, 1 \rangle$  range or  $\langle 0, 1 \rangle$  range.

- Scaling range  $\langle -1, 1 \rangle$

$$Z'(t) = \frac{2 \cdot Z(t) - (max + min)}{max - min} \quad (2.17)$$

- Scaling range  $\langle 0, 1 \rangle$

$$Z'(t) = \frac{Z(t) - \min}{\max - \min} \quad (2.18)$$

Where  $\min; \max$  corresponds to *minimum; maximum* values of the time series  $Z(t)$ .

### 2.8.5 Normalization

The general aim of normalization is an adjustment of the values by shifting and scaling, in order to obtain a so called normal distribution of the values. This produces a time series with mean property equal to 0 and standard deviation property equal to 1.

$$Z'(t) = \frac{Z(t) - \mu}{\sigma} \quad (2.19)$$

Where  $\mu$  is the mean value and  $\sigma$  is the standard deviation of the given time series.

$$\mu = \frac{1}{n} \sum_{t=1}^n Z(t) \quad \sigma = \sqrt{\frac{1}{n} \sum_{t=1}^n (Z(t) - \mu)^2} \quad (2.20)$$



## Chapter 3

# Forecasting Methods

### 3.1 Regression models

There is a lot of tasks, which require the investigation of relationships between two and more variables. Regression analysis is a typical method, that is being used for this kind of problems. The aim of regression analysis is to estimate the dependencies between main variable and a set of external factors (regressors).

The linear regression model is the simplest and the most widely used regression model. It assumes, that there is a set of external factors  $X_1(t), X_2(t), \dots, X_p(t)$ , which have an impact on the given process  $Z(t)$  and the relationship between them is linear. Forecasting model based on the linear regression is determined by an equation (2.12).

$$Z(t) = \alpha_0 + \alpha_1 X_1(t) + \alpha_2 X_2(t) + \dots + \alpha_p X_p(t) + \varepsilon_t \quad (3.1)$$

Where  $\alpha_i, i = 0 \dots p$  are regression coefficients (parameters),  $\varepsilon$  is the approximation error. In order to obtain a forecasted values  $Z(t)$  at time units  $t$ , it is necessary to have values  $X_i(t)$  at time moment  $t$ , sometimes in practice this can be impossible in some kind of problems.

The nonlinear regression models are based on assumptions, that there is given a mathematical function, that describes relationship between given process  $Z(t)$  and the external factor  $X(t)$ .

$$Z(t) = f(X(t), \alpha) + \varepsilon_t \quad (3.2)$$

While constructing the forecast model, it is necessary to determine the function parameters  $\alpha$ . For example,  $Z(t)$  dependency on  $\sin(X(t))$

$$Z(t) = \alpha_1 \sin(X(t)) + \alpha_0 + \varepsilon_t \quad (3.3)$$

In order to construct this model it is sufficient only to determine the parameters  $\alpha = (\alpha_0, \alpha_1)$ . However in practice it is not very common, that type of functional dependency between process  $Z(t)$  and external factor  $X(t)$  is already known in advance. Therefore, nonlinear regression models are used less frequently, than the linear ones.

### 3.2 Autoregressive and moving average models

Autoregressive models are based on the idea, that values of process  $Z(t)$  are linearly dependent on some number of past values of the same process  $Z(t)$ . In this model, the actual value of the process is expressed as a sum of finite linear combination of previous values and the impulses, called white noise.

$$Z(t) = c + \varphi_1.Z(t-1) + \varphi_2.Z(t-2) + \dots + \varphi_p.Z(t-p) + \varepsilon_t \quad (3.4)$$

where  $\varphi_i$  are parameters of the model;  $c$  is a constant;  $\varepsilon$  is white noise (error of the model).

The formula describes the autoregressive model of order  $p$ . This model is often denoted as  $AR(p)$ . The parameters  $c$  and  $\varphi_i$  are usually estimated by mean least squares or maximum likelihood methods.

The second model, moving average model. It plays very important role in time series description and is frequently used in relation with the autoregressive models. Moving average model of order  $q$  is described by formula:

$$Z(t) = \frac{1}{q}[Z(t-1) + Z(t-2) + \dots + Z(t-q)] + \varepsilon_t \quad (3.5)$$

where  $q$  is order of moving average and  $\varepsilon_t$  is prediction error.

In the books, moving average model of order  $q$  is usually denoted as  $MA(q)$ . Actually, moving average model is a finite impulse response filter applied to white noise.

In order to achieve better prediction quality, two previous models are often merged into one model, autoregressive and moving average model. Common model is denoted

as  $ARMA(p, q)$  and it unites a moving average filter of order  $q$  and autoregression of filtered values of order  $p$ .

If the time series data show evidence of non-stationarity, then the initial differencing step can be applied to reduce the non-stationarity. This model is usually denoted as  $ARIMA(p, d, q)$ . The parameter  $d$  represents the degree of differencing, it corresponds to the *integrated* part of the model.

Another option is an  $ARIMAX(p, d, q)$  model, that is an extension of  $ARIMA(p, d, q)$  model. It is described by formula:

$$Z(t) = AR(p) + \alpha_1 X_1(t) + \dots + \alpha_S X_S(t) \quad (3.6)$$

This model is extended by the impact of external factors. In this model, the process  $Z(t)$  is a result of model  $MA(q)$ , that are filtered values of the original process. Subsequently autoregressive forecasting, with additional regression parameters, corresponding to external factors, is performed.

### 3.3 Exponential smoothing models

Despite the fact, that Exponential smoothing methods were invented in the middle of 20th century, they are still frequently used, even today. Exponential smoothing models are widely used for modeling finance and economical processes. The basis of exponential smoothing, is an idea of repetitive revision of forecasting function, with each income of newly observed value. Exponential smoothing model assigns exponentially decreasing weights to past values, according to the age. Therefore, newly observed values have higher impact on forecasted value, than the elder ones. Functional representation of exponential smoothing model is expressed by the following equations:

$$Z(t) = S(t) + \varepsilon_t \quad (3.7)$$

$$S(t) = \alpha \cdot Z(t-1) + (1-\alpha) \cdot S(t-1) \quad (3.8)$$

$$S(1) = Z(0) \quad (3.9)$$

where  $Z(t)$  is an actual value of the time series observed at time unit  $t$ ;  $S(t)$  is a smoothed value at time  $t$ ;  $\varepsilon_t$  is an error between actual and smoothed value;  $\alpha$  is a smoothing coefficient,  $0 < \alpha < 1$ . In this model, each subsequently smoothed value  $S(t)$  is a weighted combination of previous time series value  $Z(t - 1)$  and previously smoothed value  $S(t - 1)$ .

### 3.3.1 Double exponential smoothing

Double exponential smoothing, sometimes referred as "Holt-Winters double exponential smoothing" is an improved modification of simple exponential smoothing. This model is usually used for processes, which contain a trend component. In comparison to the simple exponential smoothing, in these cases, it is necessary to deal with additional smoothing coefficient related to trend component. The model is described by the following equations.

$$S(t) = \alpha \cdot Z(t) + (1 - \alpha) \cdot (S(t - 1) + B(t - 1)) \quad (3.10)$$

$$B(t) = \beta \cdot (S(t) - S(t - 1)) + (1 - \beta) \cdot B(t - 1) \quad (3.11)$$

$$S(1) = Z(1)B(1) = Z(1) - Z(0) \quad (3.12)$$

where  $Z(t)$  is an actual value of the time series observed at time unit  $t$ ;  $S(t)$  is a smoothed value at time  $t$ ;  $\alpha$  is the data smoothing coefficient,  $0 < \alpha < 1$ ;  $\beta$  is the trend smoothing coefficient,  $0 < \beta < 1$ .

Forecasting with double exponential smoothing

In order to obtain a forecasting model based on exponential smoothing, it is necessary to have some specific amount of historical values of the given time series. The model is being built, by solving an optimization task, which consists of finding the appropriate values of  $\alpha$  and  $\beta$  parameters, such that MSE of the smoothed curve is minimal. As soon as the optimal values for parameters are estimated and the model is created, the forecasting of future values can be performed according to the following equations:

$$F(t + 1) = S(t) + B(t) \quad (3.13)$$

$$F(t + m) = S(t) + m \cdot B(t) \quad (3.14)$$

### 3.4 Artificial neural networks models

In the past few years, there can be observed a great interest in machine learning, especially in artificial neural networks sector. Artificial neural networks are tools, that are being used today for solving huge amount of tasks from different areas. The most frequent examples are time series forecasting, pattern recognition, data clustering and classification. Such a great success is determined by several reasons.

1. Artificial neural networks represent exclusively powerful tool, that enables to reproduce very complex nonlinear dependencies. For many years linear models played the leading role in the most areas, as there were a lot of well designed and optimized tools, which satisfactorily coped with assigned tasks, but problem was with tasks, for which the linear approximation is unsatisfactorily.
2. Artificial neural networks are learning from examples. Artificial neural networks receives a set of representative examples and then a learning process starts, which tries to find and extract the structure of data. Certainly, proper application of artificial neural network demands specific requirements for a correct formulation of representative data set and network's architecture. However, proper construction of a such artificial neural network allows to cope with tasks, which can be solved by the traditional algorithms only with the great difficulties. For example, pattern recognition task, practically used for face recognition, solving it in traditional way would result in a very complex problem. However, the same problem can be prospectively solved by the artificial neural networks. [6]

#### 3.4.1 Biological inspiration

Artificial neural networks are results of researches in the field of Artificial intelligence. Human brain is known to be able to deal with the problems much more complex, than the computers solve. It consist of huge number of neurons connected with each other by numerous connections. Neurons are specific nerve cells, belonging to the nervous system, that are able to distribute electrical or chemical signals. Neuron cell has a branched structure consisting of the three main parts: information inputs - dendrites, information output - axon, and the nucleus. The axon branches of the cell are connected to the dendrites of other cells with the connections called synapses. Dendrites of the

neuron receive electrical signals from other neurons through the synapses. If the total rate of the input signals, received by the dendrites of the neuron, exceeds the determined threshold, then the given neuron is going to fire an action potential. It is short-lasting process, during which the neuron sends signals to its neighbors, which also may fire. The intensity of transferred signal strongly depends on the activity of synapse between two neurons. The process of learning basically stands for an appropriate changes of the activities of the synapses connections between neurons. [7]

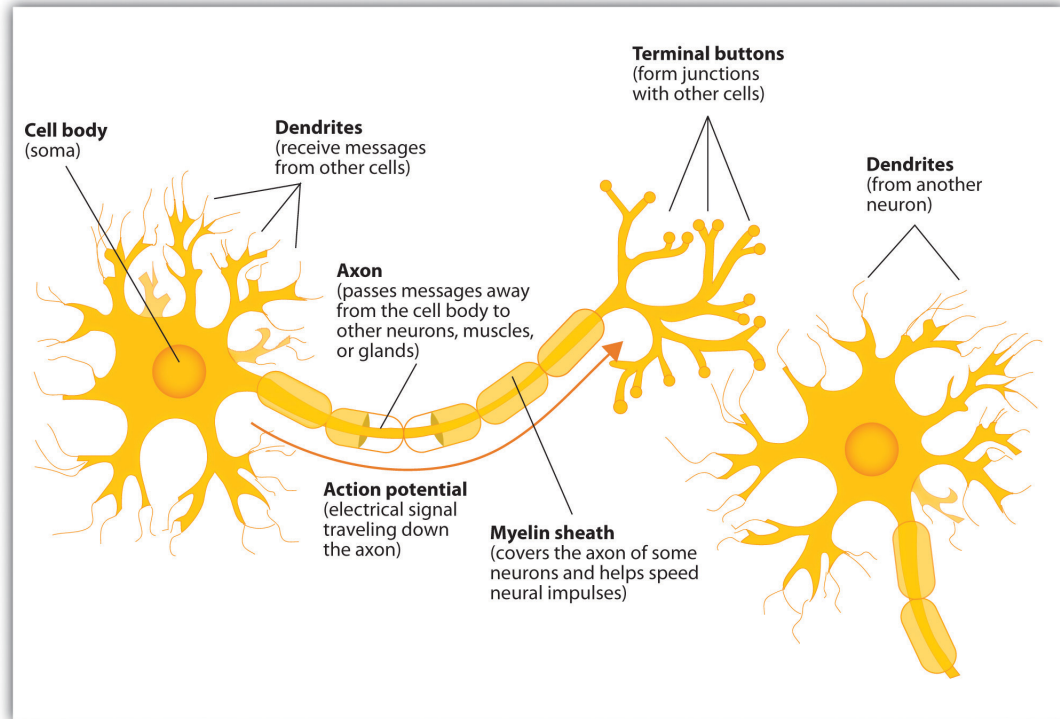


FIGURE 3.1: Biological neuron

### 3.4.2 Artificial neuron model

Artificial neuron represents a simplified model of the natural nervous cell. The evolution of artificial neurons contains several models, which have passed certain stages of development. Today, the most common artificial neuron is usually referred to the following model, determined by the three main components:

1. The set of synapses - Connecting links, each of which is characterized by its own weight. These weights correspond to the activities of synapses in biological neuron. The input signal  $x_j$ , that passes through the synapse  $j$ , which belongs to the neuron  $k$ , is multiplied by the weight  $w_{kj}$ .

2. The adder - Component, which calculates the weighted sum of signals, i.e. the linear combination. Additionally, for each neuron there is defined a threshold value  $b_k$ , denoted as "bias", which is added (or subtracted) to the weighted sum of signals. Obtained result is usually denoted as "induced local field" or "activation potential", depending on the value of  $b_k$ .
3. Activation function - Output obtained from the adder component, is passed further to the activation function. Activation function transforms the input and produces the output  $y_k$ , referred as output of neuron. [6]

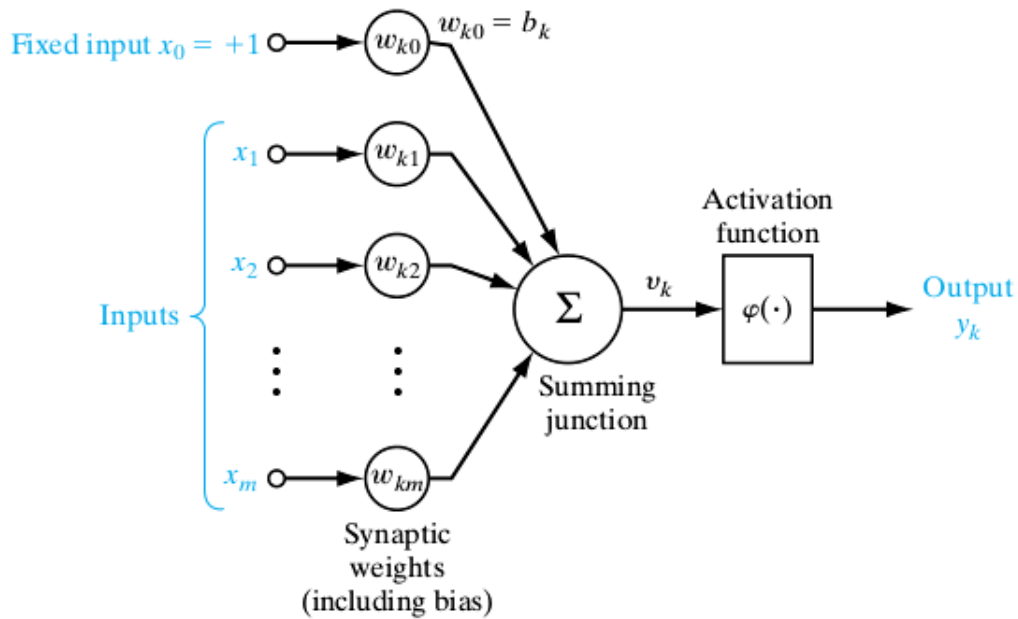


FIGURE 3.2: Artificial neuron model

In mathematical terms, artificial neuron depicted by figure 3.2 may be described by the following equations.

$$v_k = \sum_{j=0}^m w_{kj} \cdot x(j) \quad (3.15)$$

$$y_k = \varphi(v_k) \quad (3.16)$$

where  $x_0 = 1$  and  $x_1, x_2, \dots, x_m$  are the input signals;  $w_{k1}, w_{k2}, \dots, w_{km}$  are the respective synaptic weights of neuron  $k$ ;  $b_k$  is the bias;  $v_k$  is the "induced local field" or "activation potential";  $\varphi(\cdot)$  is the activation function;  $y_k$  is the output signal of the neuron. [6]

### 3.4.3 Types of Activation Function

The activation function  $\varphi(v)$ , defines the output of a neuron. There is a lot of suitable functions, that can be exploited as the activation function in artificial neurons. Appropriate selection of activation function strictly depends on the format of input and output values, and the task expected to be performed by a neural network. It is also important to mention, that the activation functions of individual neurons are not obliged to be identical, there can be easily used different activation functions inside one neural network.

The most popular activation functions [8]:

1. Threshold function - Sometimes called binary step function. Today, in practice, this activation function is used rarely. More often, it demonstrates original inspiration by the biological neuron.
2. Sigmoid function - Frequently used function, when output values are scaled in  $[0;1]$  range.
3. Hyperbolic tangent function - Similar to sigmoid function. Output values are in  $[-1;1]$  range.
4. Identity function
5. ReLU - In recent years, ReLU is becoming very popular.


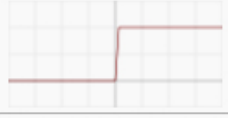
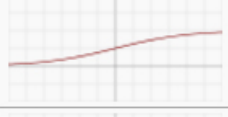
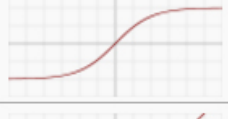
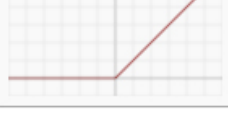
Identity		$f(x) = x$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Logistic (a.k.a. Soft step)		$f(x) = \frac{1}{1 + e^{-x}}$
Tanh		$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$

FIGURE 3.3: Activation functions



### 3.4.4 Neural Network Architectures

In the previous sections, there were described just the actions inside one artificial neuron. Now the main question is, how to connect the individual neurons to each other? In theory, neurons may be connected into neural networks with the very diverse structures. However, in practice, artificial neurons are usually grouped into layers, that later formulate a neural network.

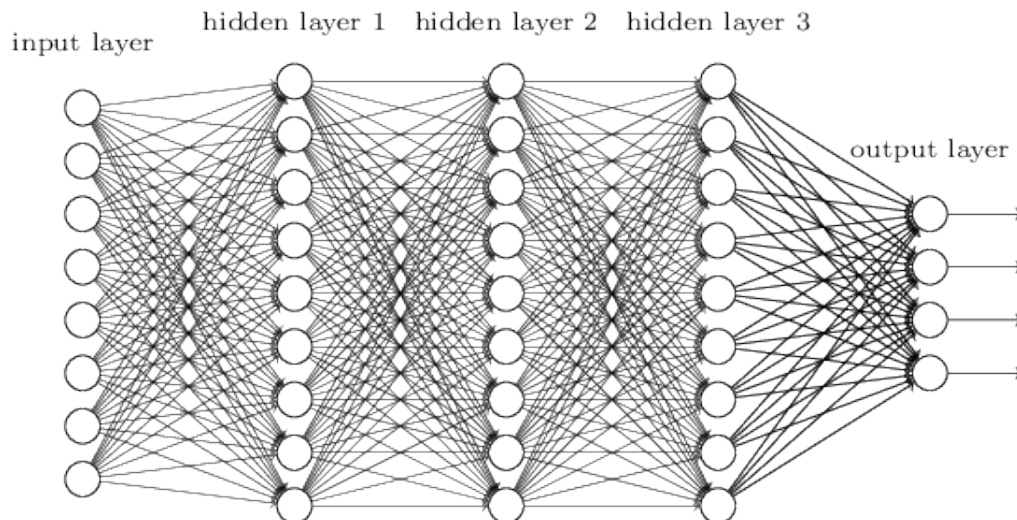


FIGURE 3.4: Artificial neural network example

Figure 3.4 demonstrates an example of neural network with one input layer, three hidden layers and one output layer. Actually, the input layer is not a real layer. It just represents the number of input values passed to the neural network. However, in the books, it is very often graphically demonstrated as the first layer of the network. All others are real layers, in sense of previously described rules. Each node in the hidden or output layer, represents a neuron. The arrows between neurons represent connections between them, and indicate the direction of signal processing. Any signal inside the network is eventually directed to the output layer, which represents an overall output of the network. All layers between the input and output layers, are called hidden layers. The name "hidden" is related to the fact, that neural network acts like a black box, and all communication with network is performed through the input and output layers, and everything, that happens inside, remains invisible to the user.

Generally, there are two main types of artificial neural networks structures:

1. Feedforward neural networks - Unites a group of networks, where the signal is passed strictly in one direction from the input layer to the output layers (Figure 3.4). Assumption is, that there is no cycles inside the network.

2. Recurrent neural networks - Represents a group of networks, which contain at least one cycle inside the network. The cycle inside the neural network means, that the output signal of some neuron, passing through the certain sequence of connections, may occur as the input to the neuron, that it has already reached. (Figure 3.5)

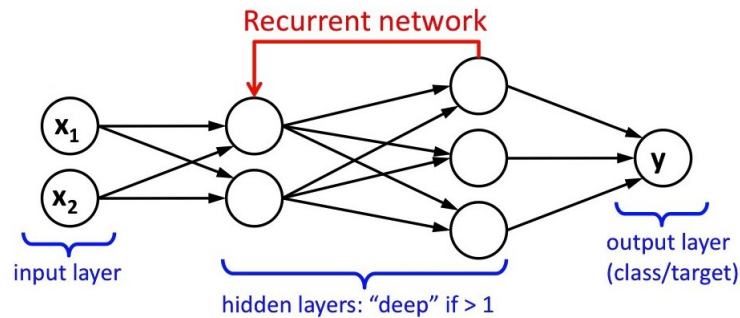


FIGURE 3.5: Artificial neural network example

Feedforward neural networks are used more frequently than recurrent networks, in part because the learning algorithms for recurrent networks are less powerful. Nevertheless, recurrent networks are still very popular. They are much closer to the biological neural networks and the idea how human brain works. Recurrent networks may be used to solve important problems, which can only be solved with great difficulty by feedforward networks. [9]

Additionally, artificial neural networks are classified as "deep" networks, if the number of hidden layers is greater than one. (Figure 3.4) [9]

### 3.4.5 Appropriate architecture

Selecting an appropriate architecture of neural network is an important step. When selecting an architecture, it is necessary to deal with following parameters:

- Number of neurons in input layer - Usually, number of neurons in input layer directly depends on the format of the input data. For example, if the neural network will be used for time series forecasting, the number of input neurons will correspond to the number historical values used for forecasting. If the neural network will be used for images classification, the number of input neurons will correspond the number of pixels of the images.
- Number of neurons in output layer - Similarly as with the number of input neurons, the number of output neurons directly depends on the performed task and the amount of output information. For example, if the neural network is used time

series forecasting, the number of output neurons will correspond to the number of forecasted values. If the neural network is used for classification of images with the handwritten number, the number of output neurons can be 10, each neuron for one number (class of images).

- Number of hidden layers - In mathematical theory, neural network with at least one hidden layer, is sufficient to approximate or learn dependencies of any non-linear function. Despite this, for many tasks it is much more suitable to use a neural network with more than one hidden layer. For more complex tasks, like images classification, are usually used deep neural networks with much more than one hidden layer. On the other hand, tasks, which do not contain so complex dependencies, they also do not require so complicated structures, as it will just lead to overfitting and decrease the performance. [9]
- Number of neurons in hidden layer - This parameter is also very sensitive to overfitting. Usually, there is no regular rule, how to choose the number neurons in hidden layer. There exist some recommendations, but the most reliable solution leads to the benchmarking. [6]

### 3.4.6 Networks training

Architecture selection is just the first step. After the neural network is constructed, it is still not ready to be exploited. During the initialization, the weights of connections between neurons are selected randomly. Before the neural network can be adequately used for required task, proper weights have to be found. This process is usually referred as a learning or training of the neural network.

There exist different learning algorithms, each of them is suitable for the specific network architecture. Backpropagation algorithm is one of the most popular training algorithms. It is very effective algorithm, but it can be used for training networks with at most one hidden layer. The majority of tasks can be easily solved by neural networks with one hidden layer, therefore backpropagation algorithm is suitable for these cases. In the case of deep networks, backpropagation leads to the vanishing gradient problem, and makes it impossible to use. [9]

### 3.4.7 Cross-validation

Before the learning process can be launched, it is necessary to perform data partitioning. Data are divided into three sets: training set, validation set and testing set. Usually, training set is the largest and it contain the data, which will be used for network training.

Validation set is used to deal with the overfitting problem. Overfitting is a common problem, which may occur when it is required to fit a model to the training data. After some moment the model perfectly fits the training set, but it will have low performance on the newly observed data.

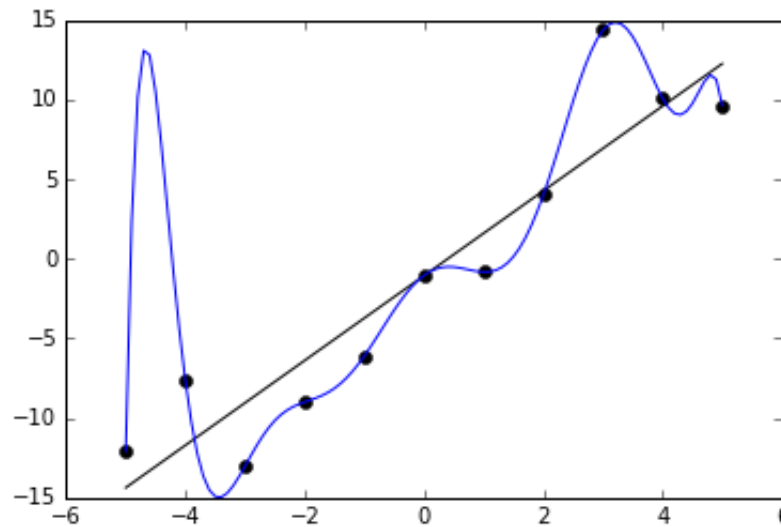


FIGURE 3.6: Overfitting example

In order to deal with the overfitting problem, the validation set is used. Neural network is trained on the training data, but the error is calculated for the validation set. Training is performed up to the moment when the error for validation set starts to increase.

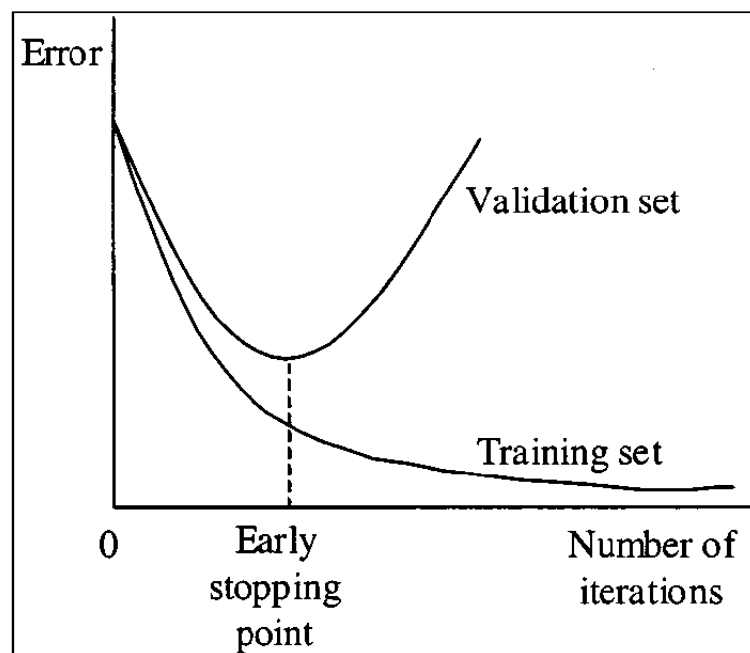


FIGURE 3.7: Cross-validation

Testing set doesn't participate in network training. It is just used to express the performance of network on the independent data.

### 3.4.8 ANN forecasting model

Artificial neural networks allow to create very powerful forecasting models. Its ability to deal with the non-linear dependencies gives a great advantage, in comparison to other forecasting methods. Before the time series data can be applied to the neural network, it is necessary to "cut" the data on the samples of the specific length, which corresponds to the number of neurons in the input layer. As well, it is required to prepare the target samples, what corresponds to the forecasted values.

As soon as the neural network is constructed and successfully trained, it represents a forecasting model and can be used for time series forecasting, as any other model.

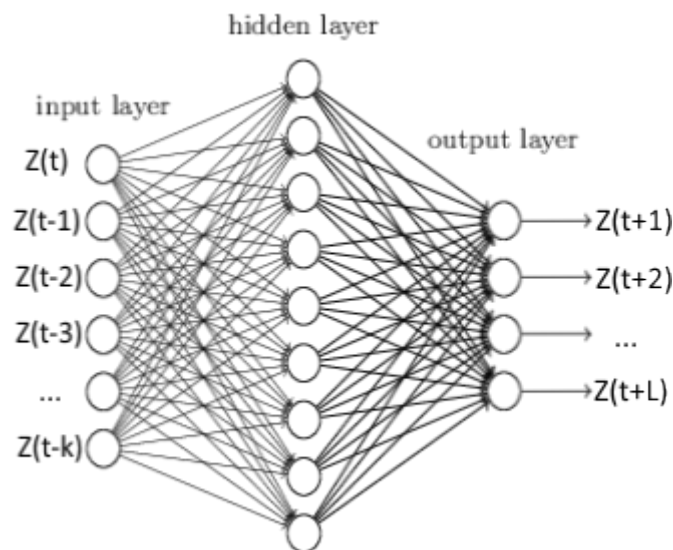


FIGURE 3.8: ANN and time series forecasting

## 3.5 Markov chain models

Forecasting models based on the Markov chains assume, that future state of the process is dependent only on its current state and is not dependent on its elder states. Markov chain models are applicable on the short-memory time series. Example of Markov chain for process with 3 states is illustrated on figure 1.3.

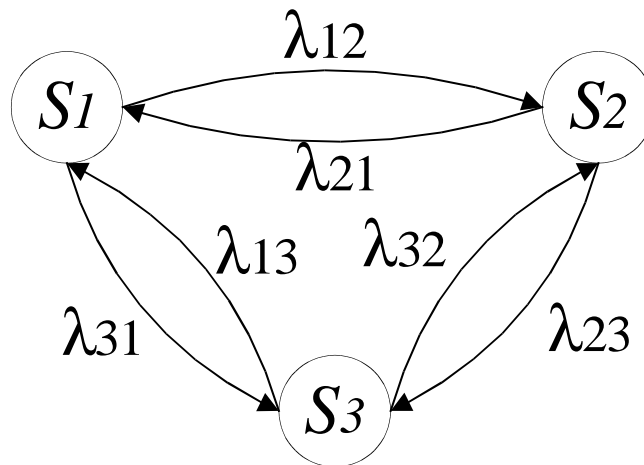


FIGURE 3.9: Markov chain model

In this model,  $S1, S2, S3$  - are states of process  $Z(t)$ ,  $\lambda_{xy}$  - probability of transition from state  $x$  to state  $y$ . By building the Markov chain model, the set of states and corresponding transitions' probabilities are defined. If the current process state is defined, the future state is selected as the state with maximal transition probability. If the transition probabilities are properly stored in matrix, subsequent future values can be determined by probability matrix's multiplication and maximum probability selection. [10]

### 3.6 Forecasting models comparison

Forecasting Model and Method	Advantages	Disadvantages
Regression models	The main advantages of the given models are: simplicity, flexibility and uniformity of calculations. Simplicity of model construction (only linear models). Transparency of all intermediate calculations.	Inefficiency and low adaptability of linear regression models for non-linear processes. Very complex non-linear model construction for the tasks with non-linear functional dependency.

Autoregressive and Moving average models	Transparency and uniformity of calculations and model's construction. Relatively not complicated model construction. The most popular frequently used forecasting method. A lot of publications and information about how to apply this method for the specific problems.	Large number of parameters required to be determined. Linearity, low adaptability and inefficiency with non-linear processes.
Artificial Neural Networks models	The main advantage of these models is a non-linearity. Neural networks can easily deal with the non-linear dependencies between future and past values of the processes. Great adaptability and scalability. Ability of parallel computations.	Large number of parameters and significant options necessary to be selected. High hardware performance requirements during the network training process. Complexity of architecture and absence of transparency.
Exponential smoothing models and methods	Transparency of intermediate calculations, simplicity and relative effectiveness. Easy model construction.	The disadvantage of this model is inflexibility.
Markov chains models	Transparency of intermediate calculations.	Impossibility of long term forecasting.

## Part II

# Practical part



## Chapter 4

# Internet traffic data forecasting

The main tasks of this thesis are: analysis of provided time series data sets and development of the corresponding forecasting models. All data sets that have been provided to me, contain local outdoor temperature values of the specific buildings. Prediction of this kind of temperature values represents a real practical task and plays an important role for their further application by the thermal control units. In order to solve this task, the practical part of thesis has been performed in two steps:

1. In the first step, there will be tested three different methods for time series forecasting: Autoregressive-Moving-average methods, methods based on artificial neural networks and Exponential smoothing method. Their effectiveness will be compared on the public data sets downloaded from the internet. For this purpose, there have been selected two data sets from absolutely different fields of activity. The results will be summarized and proposition about forecasting methods for the next step will be made.
2. In the second step, the main task of the thesis will be solved. Based on the experience, obtained from the previous experiments, there will be developed forecasting models for each of provided data sets with temperatures values.

### 4.1 Internet traffic data set experiments

In the following section will be used an Internet traffic data set obtained from the following source: <https://datamarket.com/data/set/232h/>

Data set represents aggregated traffic values (in bits) of an academic network backbone in the United Kingdom. The values were collected during the period from 19 November

2004 to 27 January 2005 with one hour interval. The following Figure 4.1 demonstrates plotted graph of the given data set.

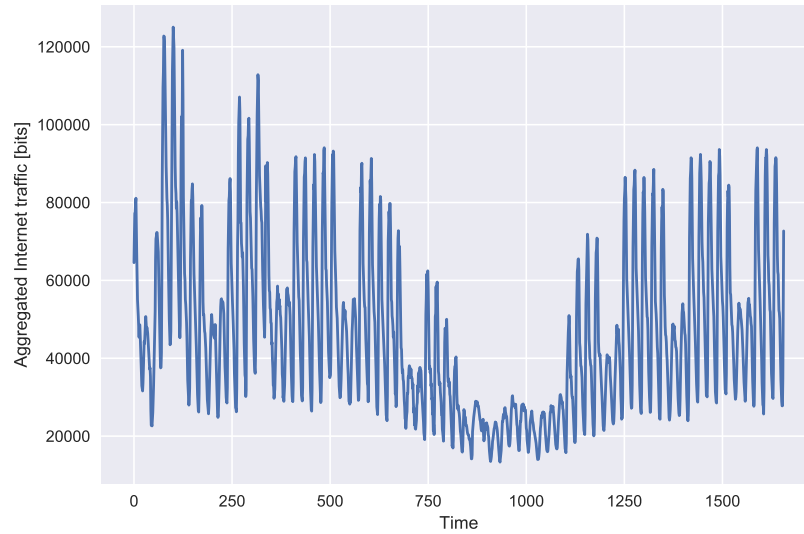


FIGURE 4.1: Plot of the internet traffic data (in bits)

## 4.2 Data analysis and preprocessing

The initial and the most important step in time series analysis is determination of whether a time series is stationary or not. This step is important because the most of the forecasting methods can deal only with stationary time series. In the theoretical part of the thesis it was described, that stationary time series is one, whose statistical properties like mean and variance do not depend on time, at which the series is observed. From the practical point of view, the time series non-stationarity is usually caused by the presence of the trend or seasonality components inside the series.

Very often, simple visual observation of graphs of rolling mean and rolling variance functions helps to make suggestion, whether the series is stationary or not [11]. Both functions belong to so called "rolling" analysis of the time series, when the sliding window technique is used to plot the progress of statistical parameters for the given size of window. The plot of rolling mean and rolling variance functions (Figure 4.2) show the obvious evidence of the series non-stationarity. Definitely, the mean property do not represent a constant progress over time, as well as the variance demonstrates the regular fluctuations.

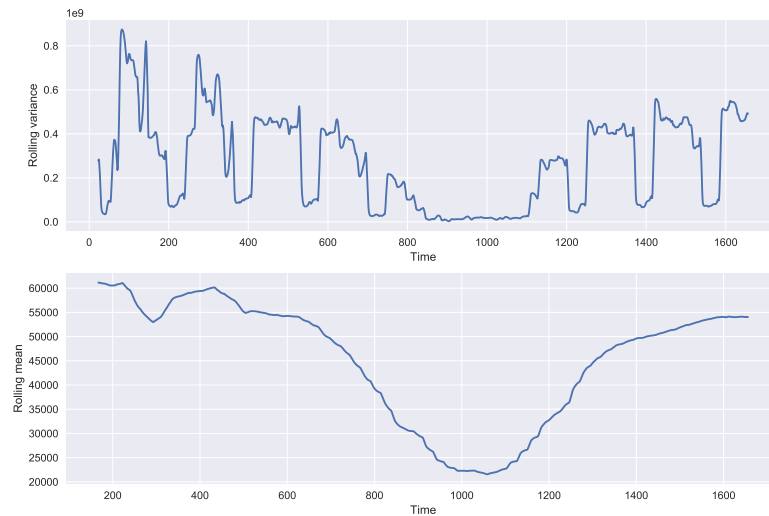


FIGURE 4.2: Plot of the internet traffic data (in bits)

Analysis of ACF and PACF plots is another useful and very informative method for identifying time series non-stationarity. There exist general rules, how to interpret the results of ACF a PACF functions. It is known, that for a stationary time series, the ACF drops to zero relatively quickly, while the ACF of non-stationary time series decreases slowly [12].

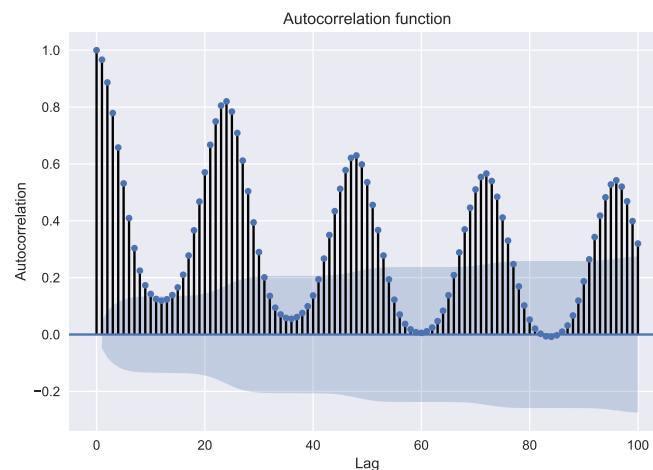


FIGURE 4.3: The plot of ACF of raw time series data.

The plot of ACF (Figure 4.3) demonstrates slowly decreasing progress with clear evidence of regularly repeating patterns after every 24 values. This corresponds to the non-stationarity of the series, with the presence of seasonality and trend components.

Finally, there has been used an ADF test (Augmented Dickey-Fuller test) to confirm the previous assumptions about the series [13]. This is one of the statistical unit root tests, that is frequently used for determination of series stationarity. The null-hypothesis for

an ADF test is that the data are non-stationary. Usually 5% threshold is being used, what means, that null-hypothesis is rejected if the  $p$ -value is less than 0.05. The result of the ADF test for the raw data of the given data set confirms the assumption of time series non-stationarity, the  $p$ -value = 0.11617, what is greater than 0.05.

Now, there is definitely no doubts, that the time series is non-stationary and that it requires some preprocessing steps to stationarize it. At the beginning, the log transformation has been performed. This helped to stabilize the variance of the series, but it wasn't enough to make it stationary [14]. Another important transformation is differencing. It stabilizes the mean of the series and eliminates the trend and seasonality components. There are two types of differencing, that should be considered for the given time series, the seasonal and non-seasonal one. At first, the non-seasonal differencing has been performed, it eliminated the trend from the series, but there still remained seasonal patterns in the plot of ACF, repeating after every 24 hours. Differencing should be performed one more time. It is known, that simple non-seasonal differencing can not deal with strong seasonality effect. Therefore, after one non-seasonal differencing, there will be also performed one seasonal differencing with  $lag = 24$  [15]. The ACF and PACF plots of the time series after all transformations are demonstrated in the Figure 4.4 and Figure 4.5.

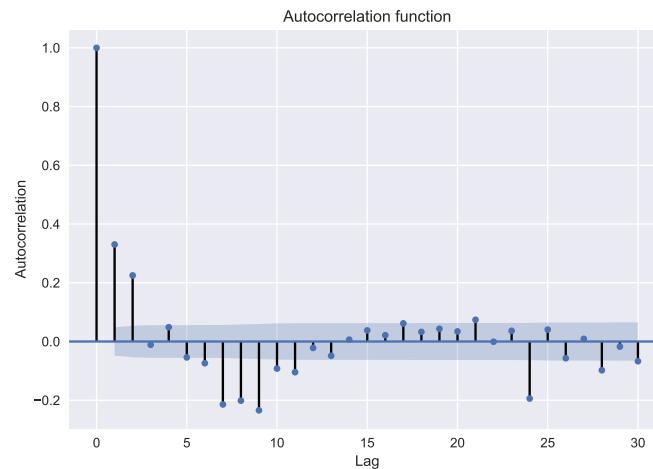


FIGURE 4.4: The ACF plot of internet traffic data after log transformation, one non-seasonal differencing and one non-seasonal differencing with lag=24.

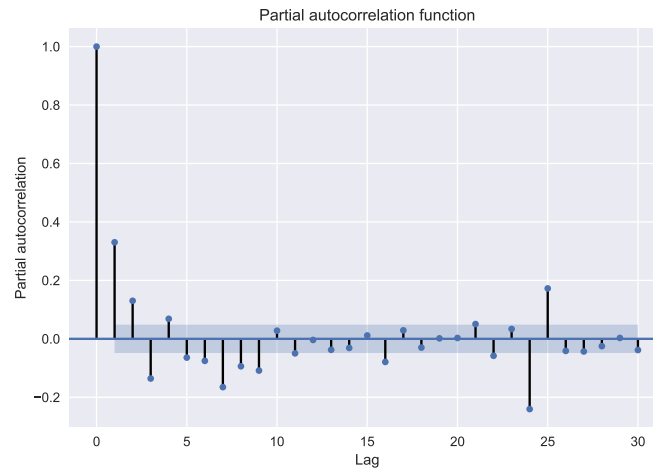


FIGURE 4.5: The PACF plot of internet traffic data after log transformation, one non-seasonal differencing and one non-seasonal differencing with lag=24.

The ACF and PACF plots of the preprocessed time series data demonstrate signs of stationarity. This is also confirmed by the ADF test. The p-values of the ADF test of preprocessed data is far less than the critical value 0.05 and assumption about time series stationarity is confirmed. At this moment, the time series is ready to be applied by forecasting methods.

### 4.3 Autoregressive-Moving-average method

The forecasting models, that will be developed in the following section are based on the analysis performed in the section 4.2. At the beginning, the data set is divided into training data 85% and testing data 15%. The next step is to analyze the results of ACF and PACF plots and make suggestion about the autoregressive and moving average parameters of the model. There exist some general rules, how to identify the parameters [16].

The analysis in the previous section demonstrated the necessity of one seasonal and one non-seasonal differencing. This suggests the use of  $SARIMA(p, d, q) \times (P, D, Q)$  model, where both differencing parameters  $d$  and  $D$  are equal to 1. This corresponds to the use of one non-seasonal and one seasonal differencing. The rest of parameters are going to be selected based on the analysis of ACF and PACF. In the Figure 4.4 there can be observed, that plot of ACF tails off after the lag 11 and the plot of PACF tails off after the lag 9. This suggests, that autoregressive parameter  $p$  has to be tested up to 9 and moving average parameter  $q$  has to be checked up to 11. In the plot of ACF there can be also observed significant negative peak at lag 24, what corresponds to the effect of seasonal component. According to the referenced rules, this should be solved by adding

seasonal moving average parameter to model. Each model will be trained on training data set and then its performance will be tested on test data set. To choose the optimal SARIMA model, the MSE rate will be used.

Experimentally it has been tested, that  $SARIMA(9, 1, 8) \times (0, 1, 1)$  model performs better than other models. The corresponding MAPE rate is 4.53%. Further increasing of the parameters was pointless and didn't lead to improvement of performance. This is can be explained by effect of overfitting. Figure 4.6 demonstrates the continuous plots of 1-hour ahead forecasted values and the actual values of the test data set.

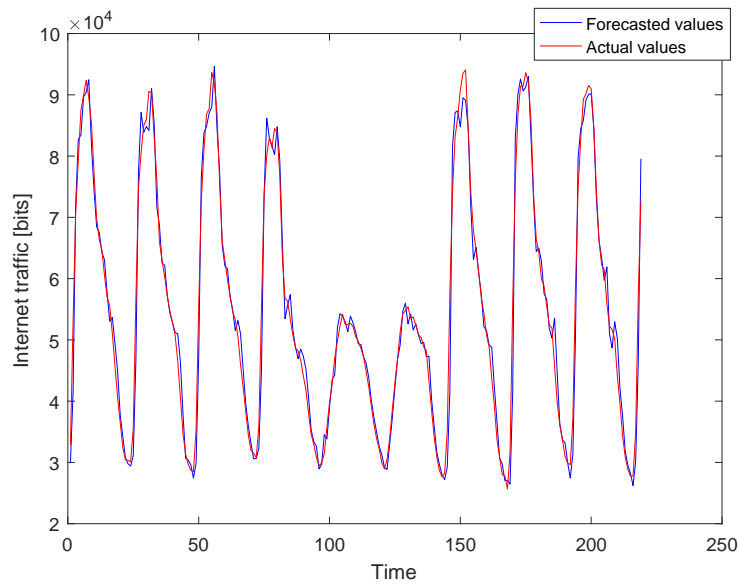


FIGURE 4.6: The plots of one hour ahead forecasted values (blue) and actual values (red) of the test data set.

The following Table 4.1 demonstrates MAPE and RMSE rates of  $SARIMA(9, 1, 8) \times (0, 1, 1)$  model for different forecast horizons.

	MAPE	RMSE
1-hour ahead forecasts	4.52%	2343.9
2-hour ahead forecasts	7.17%	6040.6
3-hour ahead forecasts	9.79%	10463.0

TABLE 4.1: Forecasting performance of  $SARIMA(9, 1, 8) \times (0, 1, 1)$  model for different forecast horizons.

## 4.4 Artificial neural networks method

In the following section, the artificial neural networks will be applied to the forecasting of internet traffic data. The initial raw data set presented in chapter demonstrated the obvious signs of non-stationarity. In theory, multilayer neural networks are able to fit any even non-stationary time series data [9]. Although, in practice, it is highly recommended to stationarize the time series before it is applied to neural networks, because proper data preprocessing steps significantly accelerate the network training process. In addition to the preprocessing transformations described in section 4.2, there are two more transformations required to be performed: data normalization and creation of the input and target data sets for network training.

For the purposes of the given experiment, it was decided to use the neural networks with multiple hidden LSTM layers and the feedforward output layer. According to the publication [17], this kind of architecture should be suitable for time series forecasting problem. Now the whole complexity of the experiment is based in tuning of the hyperparameters of the neural network. For this purposes, the heuristic described in the following book has been used [9].

Hyperbolic tangent activation function has been used for hidden LSTM layers and linear activation function has been used for output feedforward layer. Different training algorithms have been tested and finally the RMSProp training algorithm with learning rate  $\text{Alpha} = 0.001$  and mini-batch size equal to 5 has been selected. The cross validation technique has been used to prevent the overfitting and stop the training process. At this point it is assumed, that time series data have passed all necessary transformations and are ready to be used. The inputs to the network are presented as the vectors with 24 timesteps, what corresponds to historical values for the one last day. The number of hidden layers and the corresponding number of neurons have been experimentally adjusted. Two hidden LSTM layers with the corresponding number of neurons 50 and 10 demonstrated the lowest forecast error. The given forecasting model has been tested for three different forecast horizons: 1, 2 and 3 hours ahead.

Figure 4.7 demonstrates the diagram of the previously described network.

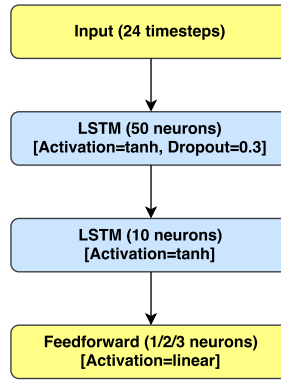


FIGURE 4.7: Diagram of ANN used in this experiment. It demonstrates structure of network, number of neurons, activation functions and dropout regularization parameters of individual layers.

Figure 4.8 demonstrates the continuous plots of 1-hour ahead forecasted values and the actual values of the test data set.

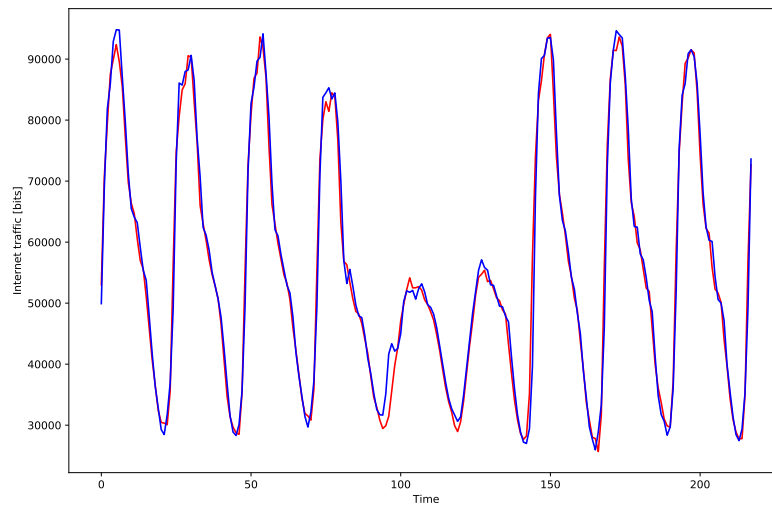


FIGURE 4.8: The plots of one hour ahead forecasted values (blue) and actual values (red) of the test data set.

The following Table 4.2 demonstrates MAPE and RMSE rates of ANN model for different forecast horizons.

	MAPE	RMSE
1-hour ahead forecasts	3.50%	1804.8
2-hour ahead forecasts	4.86%	2882.7
3-hour ahead forecasts	5.63%	3410.9

TABLE 4.2: Forecasting performance of ANN model for different forecast horizons.



## 4.5 Exponential smoothing method

In this section, double exponential smoothing, also referred as Holt-Winters method, will be used for time series forecasting. Estimation of the forecasting model is basically the process of selecting the optimal Alpha and Beta parameters, such that the MSE rate on the training set is minimal. For the given time series data the following parameters Alpha=0.95 and Beta=0.025 demonstrated the lowest RMSE rate on the training data. Figure 4.9 demonstrates the continuous plots of 1-hour ahead forecasted values and the actual values of the test data set.

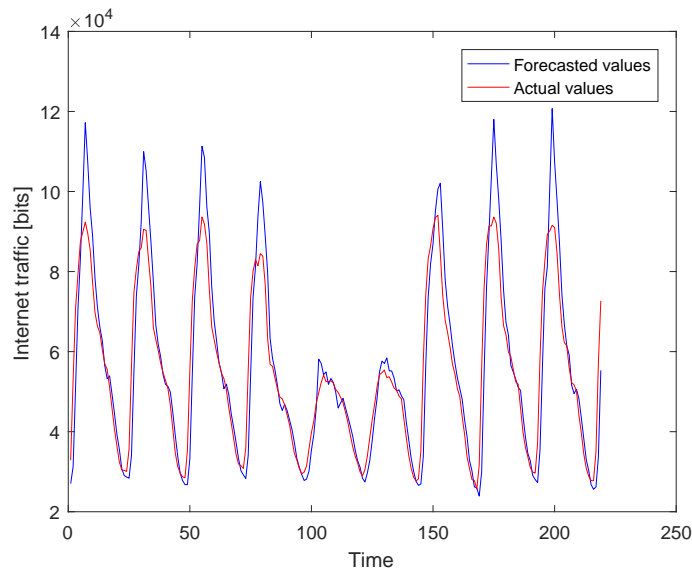


FIGURE 4.9: The plots of one hour ahead forecasted values (blue) and actual values (red) of the test data set.

The following Table 4.3 demonstrates MAPE and RMSE rates of Double exponential smoothing model for different forecast horizons.

	MAPE	RMSE
1-hour ahead forecasts	10.23%	6182
2-hour ahead forecasts	15.36%	14352
3-hour ahead forecasts	20.64%	24694

TABLE 4.3: Forecasting performance of Double exponential smoothing model for different forecast horizons.

## 4.6 Experiments summary

The main aim of the experiments in this chapter, is to demonstrate the effectiveness of time series forecasting methods and corresponding preprocessing transformations presented in the theoretical part of the thesis. For this purpose, there has been used publicly available data set of the internet traffic data. The given data set represents a typical time series data from practice. It contains both, the trend and the seasonality components, what allows to test all required preprocessing transformations.

Afterwards, three different time series forecasting methods have been tested. The ANN model presented the best forecasting performance, but all three methods demonstrated the remarkable results and confirmed, that they can be adequately used for time series forecasting.

## Chapter 5

# Main experiments

The forecasting methods demonstrated in the previous sections, as well as the time series analysis methods, have proven themselves in the experiments on the public data set of internet traffic data. Now, the given methods will be applied for solving the main task of the thesis – forecasting of the local outdoor temperature. The main aim of the task is to develop the qualitative forecasting models such, that they can be further integrated to other applications and provide adequate temperature forecasts for different control units.

Temperature data set corresponds to the individual building and contains two columns of values, more precisely, two individual time series. The first one, has been already mentioned, it represents the local outdoor temperature. The second one is a so called "equitherm outdoor temperature". This series represents a 3-hour forecasts obtained from the meteorological station, adjusted for a difference between the local temperature and provided forecast, observed in past. This technique is currently used as a simple forecasting method and its forecasting performance will serve as a benchmark value for development of more complex forecasting models. The values of "equitherm outdoor temperature" series will be used for backward computation of pure meteorological forecast, that will be further used as an external factor for extension of the forecasting models, in order to improve the forecasting performance.

The structure of the section is as follows. At the beginning, data analysis will be performed and data will be preprocessed in order to guarantee their stationarity. Afterwards, forecasting methods, that have been tested on the public data set, will be used to create forecasting models without external factors. In further steps, the models will be extended in order to increase forecasting performance.

## 5.1 Data analysis and preprocessing

Data set represents local outdoor temperatures in degrees Celsius during the period from 20.12.2015 to 30.4.2016. Records are made at ten minute intervals, producing 19146 values in total. The Figure 5.1 demonstrates the graph of raw data.

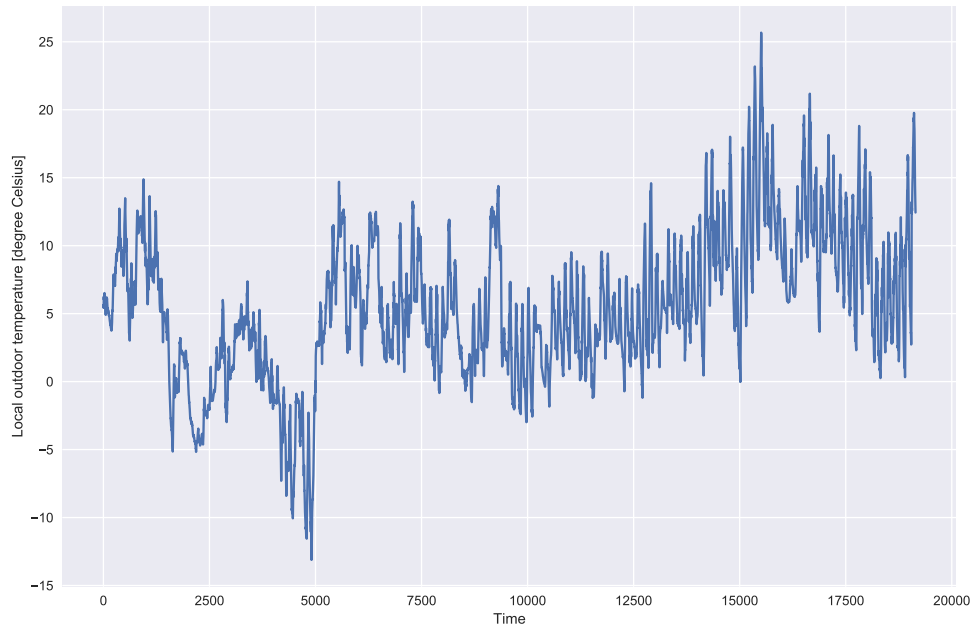


FIGURE 5.1: Plot of the local outdoor temperature data (in degree Celsius)

Figure 5.2 demonstrates the plot of ACF of raw data. Very slow decrease of ACF tells about the obvious evidence of time series non-stationarity. Additionally, the ADF test has been performed. The p-value of the test is equal to 0.1136 , what is greater than 0.05. This means, that the null-hypothesis of the test can be rejected and the assumption about time series non-stationarity is confirmed.

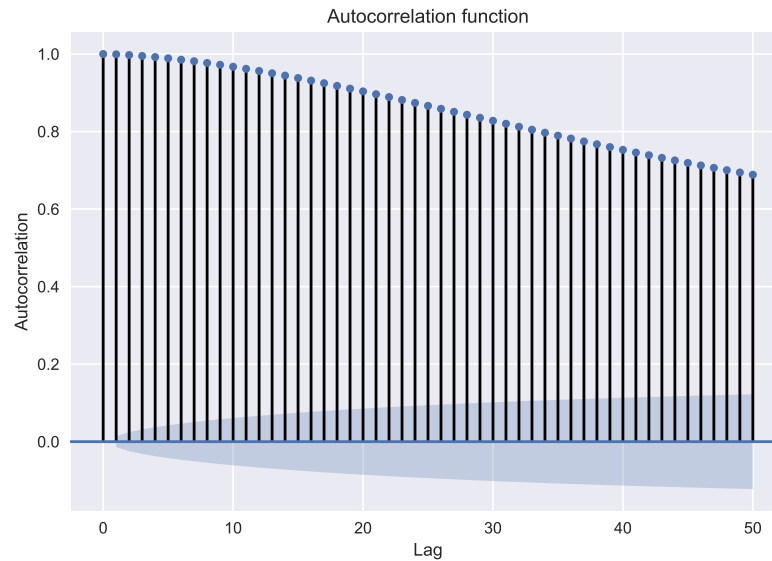


FIGURE 5.2: ACF plot of the local outdoor temperature data

The non-stationarity of the series can be easily explained. It's seasonality component reflects the daily temperature fluctuations and the trend is a consequence of temperature changes among the year. In order to stationarize the series, the following preprocessing transformations have been performed. Firstly, the simple non-seasonal differencing is used, this allows to remove the trend. The plot of ACF after one non-seasonal differencing is demonstrated in the Figure 5.3.

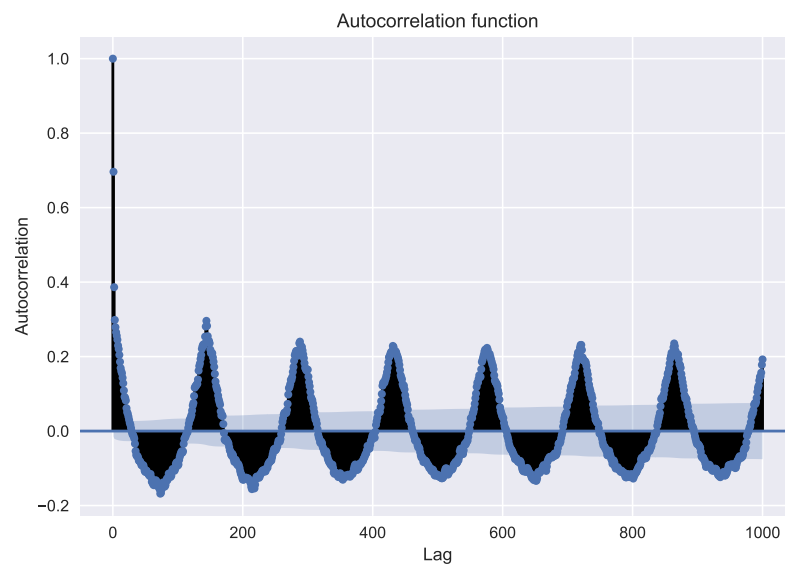


FIGURE 5.3: ACF plot of the local outdoor temperature data after one non-seasonal differencing

Now, the ACF decreases much faster, but still there can be observed patterns repeating after every 144 values, this corresponds to the 24 hour periods. Simple differencing didn't

manage to remove the seasonality component. Therefore, the seasonal differencing with  $lag = 144$  should be also used [15].

The last data preprocessing step is detection of the outliers. In the plot of differenced data there still can be observed some sharp and suspicious peaks. Data outliers don't describe the typical series behavior and their presence in the training sets is not recommended. For this purposes the Hampel filter has been used [5]. This is one of the modifications of the median filter, which is used to check the values, and replace them if they lie far enough from the median, but this filter should be used only on the training data. The plot of preprocessed data after all transformations is demonstrated in Figure 5.4.

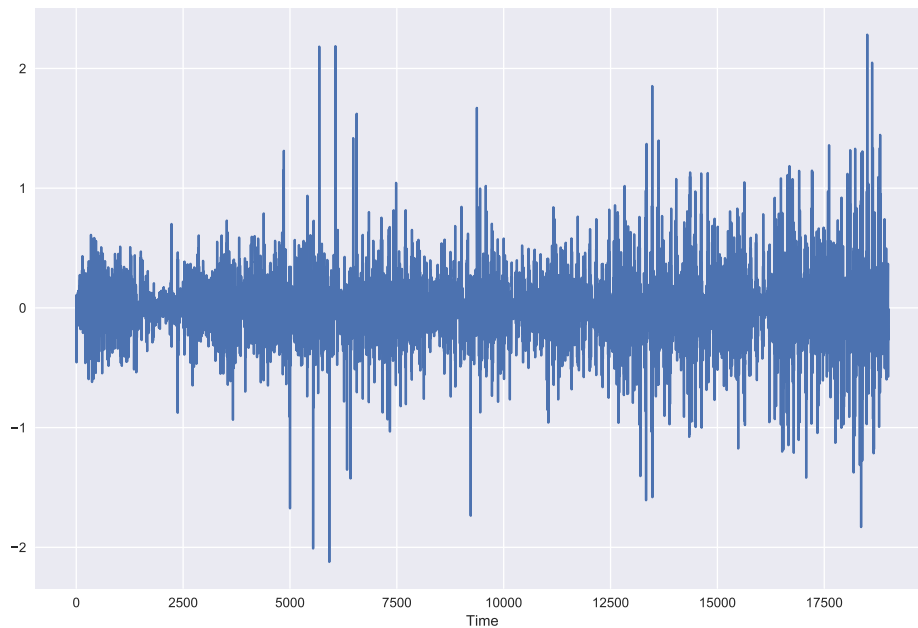


FIGURE 5.4: The plot of preprocessed local outdoor temperature data

Finally, the ADF test has been used again to confirm the stationarity of the preprocessed data. The p-values of the test is far less then the critical value 0.05 and assumption about time series stationarity is confirmed. The last data preparation step is division of the time series data into the training data 85% and testing data 15%.

## 5.2 Autoregressive-Moving-average method

Data analysis demonstrated the clear presence of the seasonal component in the data set, what causes the necessity of one seasonal and one non-seasonal differencing. This suggests the use of  $SARIMA(p, d, q) \times (P, D, Q)$  model, where both differencing parameters  $d$  and  $D$  are equal to 1. Basically, the SARIMA model is just an extended version

of ARIMA model, with the ability to deal with the seasonal data. Now, the plots of ACF and PACF of the processed data should be analyzed, to make an initial suggestions about the rest of the parameters of the given model.

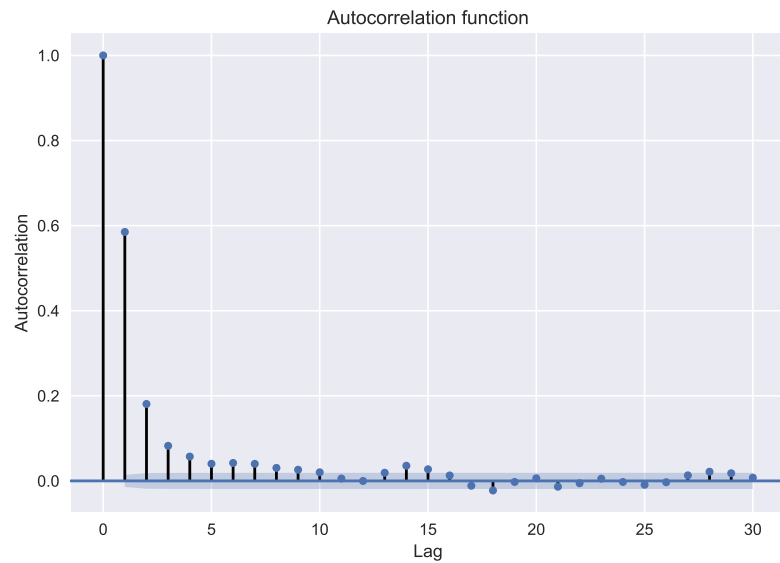


FIGURE 5.5: ACF plot of preprocessed local outdoor temperature data

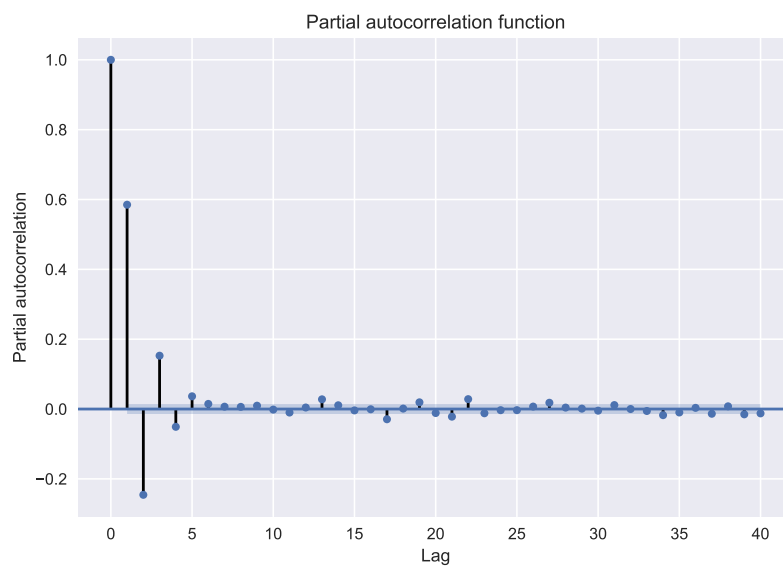


FIGURE 5.6: PACF plot of preprocessed local outdoor temperature data

Plot of ACF (Figure 5.5) tails off after the lag 15 and the plot of PACF (Figure 5.6) tails off after the lag 21. According to the known rules [16], this suggests, that autoregressive parameter  $p$  has to be tested up to 21 and moving average parameter  $q$  has to be tested up to 15. The seasonal parameters of the model should be selected according to the correlation value for the first periodic lag. In this case, correlation value is negative, and

according to the referenced rules [16], this considers the use of seasonal moving average component.

To select the optimal SARIMA model, the MSE criterion has been used. Experimentally, it has been tested, that  $SARIMA(9, 1, 5) \times (0, 1, 1)$  model performs better than other models, its corresponding MAPE measure is 6.25% for 1-hour forecast horizon. Due to the overfitting, further increasing of the parameters was pointless and didn't lead to improvement of performance. Table 5.1 demonstrates the effectiveness of  $SARIMA(9, 1, 5) \times (0, 1, 1)$  model for different forecast horizons.

	MAPE	RMSE
1-hour ahead forecasts	6.25%	1.31
2-hour ahead forecasts	10.39%	3.15
3-hour ahead forecasts	14.21%	5.41

TABLE 5.1: Forecasting performance of  $SARIMA(9, 1, 5) \times (0, 1, 1)$  model for different forecast horizons.

### 5.2.1 Forecasting with external factors

In this section, already known  $SARIMA(9, 1, 5)$  model will be extended by the effect of external factors. This kind of models is usually denoted as SARIMAX, where the “X” letter stands for the external factor.[18]

Figure 5.7 demonstrates the plots of values forecasted by  $ARIMAX(9, 1, 5) \times (0, 1, 1)$  model with 1-hour forecast horizon and actual values. The plots are demonstrated for three days period of the test data.



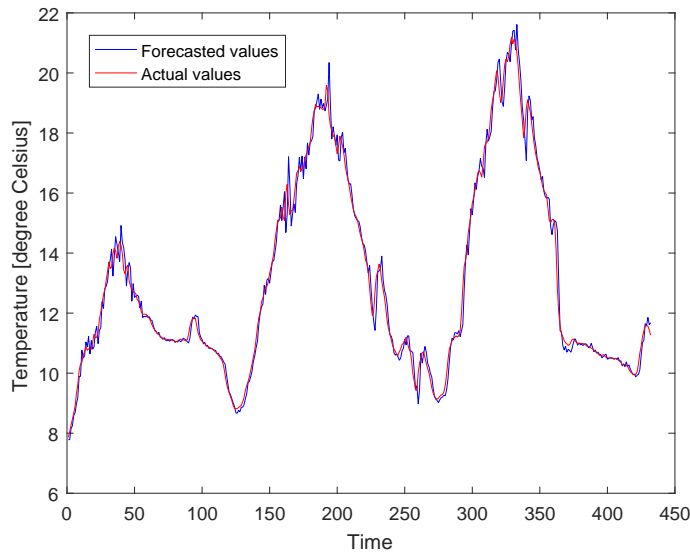


FIGURE 5.7: The plots of one hour ahead forecasted values (blue) and actual values (red) of the test data set.

Table 5.2 demonstrates the effectiveness of  $SARIMAX(9, 1, 5) \times (0, 1, 1)$  model for different forecast horizons.

	MAPE	RMSE
1-hour ahead forecasts	4.70%	0.79
2-hour ahead forecasts	7.69%	1.87
3-hour ahead forecasts	11.04%	3.50

TABLE 5.2: Forecasting performance of  $SARIMAX(9, 1, 5) \times (0, 1, 1)$  model for different forecast horizons.

### 5.3 Exponential smoothing method

In this section, double exponential smoothing, also referred as Holt-Winters method, will be used for time series forecasting. As it was described in the theoretical part of thesis, exponential smoothing is relatively simple but powerful tool. The main disadvantage of the given method is, that it can't take into account the effect of the external factors.

Estimation of the forecasting model is basically the process of selecting the optimal Alpha and Beta parameters, such that the RMSE error on the training set is minimal. For the given time series of the local outdoor temperature the following parameters Alpha = 0.995 and Beta = 0.01 demonstrated the lowest RMSE error on the training data.

Figure 5.8 demonstrates the plots of values forecasted by double exponential smoothing model with 1-hour forecast horizon and actual values. The plots are demonstrated for three days period of the test data.

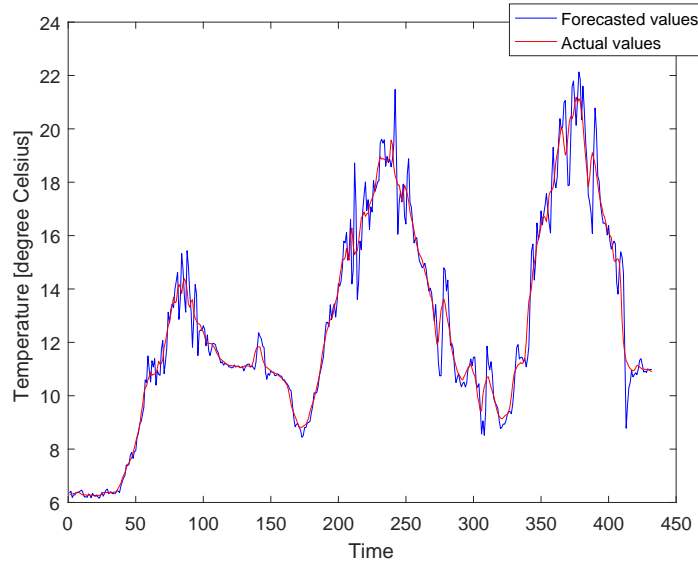


FIGURE 5.8: The plots of one hour ahead forecasted values (blue) and actual values (red) of the test data set.

Table 5.3 demonstrates the effectiveness of double exponential smoothing model for different forecast horizons.

	MAPE	RMSE
1-hour ahead forecasts	8.30%	1.78
2-hour ahead forecasts	15.85%	4.81
3-hour ahead forecasts	23.69%	8.78

TABLE 5.3: Forecasting performance of double exponential smoothing model for different forecast horizons.

## 5.4 Artificial neural networks method

In this thesis, all experiments with the neural networks will be performed in Python programming language using the Keras library and Theano backend. Keras library provides wide offer of different layers, training algorithms and other useful tools for deep learning tasks. For the purposes of this thesis, it was decided to use the LSTM networks, which belong to the class of recurrent neural networks. According to the numerous researches, LSTM networks and RNNs in general are a good choice for sequential tasks,

like speech recognition, handwriting recognition and time series forecasting as well, where each new output of the network is dependent on its previous outputs.

The whole complexity of the experiments with ANNs is based in tuning of the hyper-parameters of the network. There should be selected parameters like number of layers, number neurons in the layers, type of activation functions, the batch size and many others. This task is especially hard, when the neural network is used for some new problem. Sometimes, it may seem, that neural networks can not deal with the given problem, but usually it is because of the incorrectly selected parameters. For the smaller networks the grid search technique can be successfully used for parameters selection, but with the increasing number of layers the number of combinations to be tested is exponentially increased, what results in a very long lasting process, especially when training data set contains large number of instances. Therefore, different heuristic techniques are used. One of them is described in book [9] and will be used in the experiments of this thesis. Its general idea is based on the following steps:

- reduce the training set in order to speed up experimentation
- find some initial architecture, that will demonstrate at least trivial learning progress
- adjust the parameters to improve the performance on the validation set
- return to the initial training set and perform final optimizations

#### 5.4.1 Forecasting without external factors

At this moment it is considered, that time series data, have passed all necessary pre-processing steps, described in section 5.1. Additionally, it is highly recommended to perform data scaling, before they are applied to neural networks. The required range of scaling usually depends on the activation function used in the first layer of the network. In this case, hyperbolic tangent function is going to be used in the first LSTM layer and  $(-1; 1)$  scaling range is selected.

For the purposes of this experiment, it was decided to use the neural networks with multiple hidden LSTM layers and the feedforward output layer. Hyperbolic tangent activation function has been used for hidden LSTM layers and linear activation function has been used for output feedforward layer. Different training algorithms have been tested and finally the RMSProp training algorithm with learning rate  $\text{Alpha} = 0.0005$  and batch size equal to 5 has been selected, as it demonstrated the most stable training

ability for the given problem. The cross validation technique has been used to prevent the overfitting and stop the training process.

The inputs to the network are presented as the vectors with 60 timesteps, what corresponds to historical values for the last 10 hours. The number of hidden layers and the corresponding number of neurons has been experimentally adjusted. Two hidden LSTM layers with the corresponding number of neurons 200 and 50 demonstrated the lowest forecast error. The given forecasting model has been tested for three different forecast horizons: 6, 12 and 18 steps ahead, what corresponds to the forecasting of temperature values for 1, 2 and 3 hours ahead. Figure 5.9 demonstrates the diagram of the previously described network.

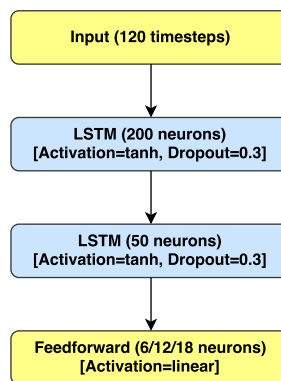


FIGURE 5.9: Diagram of ANN used in this experiment. It demonstrates structure of network, number of neurons, activation functions and dropout regularization parameters of individual layers.

Table 5.4 demonstrates the effectiveness of ANN model for different forecast horizons.

	MAPE	RMSE
1-hour ahead forecasts	3.84%	0.55
2-hour ahead forecasts	6.98%	1.58
3-hour ahead forecasts	12.32%	4.23

TABLE 5.4: Forecasting performance of ANN model for different forecast horizons.

## 5.5 Forecasting with external factors

In this task it is necessary to forecast the local outdoor temperature. It is generally known, that forecasting of the meteorological time series belong to the hard tasks. The problem is, that pure mathematical forecasting of the temperature doesn't lead to the qualitative results for longer forecast horizons. Therefore, the ANN forecasting model from the previous section will be extended by adding an external factors. In this case, the

meteorological forecast from the nearest meteorological station will be used. It is sure, that meteorological forecast and real local outdoor temperatures will differ at many cases, because the local outdoor temperature is affected by many other local factors, that aren't taken into account by meteorological forecast, but nevertheless the general tendencies will remain the same.

The initial LSTM network, presented in the previous section will be extended by adding another input vector, containing the meteorological 3-hour forecast values. It is also necessary to mention, that external factor time series should undergo similar preprocessing steps as the original time series, in order to keep the same scale of input data.

Now, after the new input vector has been added, it is also necessary to tune again the hyper-parameters of the network. Parameters like training algorithm, learning rate and batch size remained unchanged, but the number of neurons in the network should be increased, as the total amount of information fed to the neural network is now larger. The number of neurons in LSTM hidden layers has been experimentally selected to be 250 in the first hidden layer and 75 in the second hidden layer. Figure 5.10 demonstrates the diagram of the previously described network.

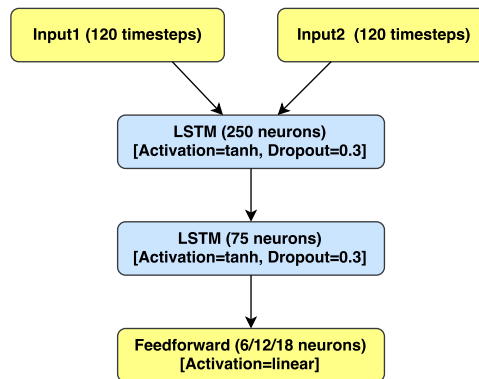


FIGURE 5.10: Diagram of ANN used in this experiment. It demonstrates structure of network, number of neurons, activation functions and dropout regularization parameters of individual layers.

Figure 5.11 demonstrates the plots of values forecasted by ANN model with external factor for 1-hour forecast horizon and actual values. The plots are demonstrated for three days period of the test data.

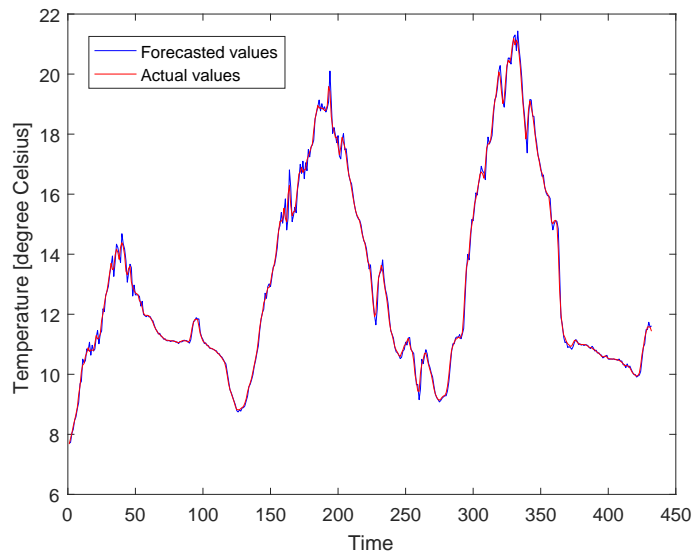


FIGURE 5.11: The plots of one hour ahead forecasted values (blue) and actual values (red) of the test data set.

Table 5.5 demonstrates the effectiveness of ANN model for different forecast horizons, after adding an external factor.

	MAPE	RMSE
1-hour ahead forecasts	2.91%	0.33
2-hour ahead forecasts	5.49%	1.04
3-hour ahead forecasts	7.01%	1.61

TABLE 5.5: Forecasting performance of ANN model for different forecast horizons, after adding an external factor

## 5.6 Data set extension

One of the main problems that may occur in training of neural networks is overfitting. Especially, the larger networks with multiple hidden layers are prone to this. There are more options how to deal with this problem. The most intuitive and useful one is to extend the training data set by adding new training data. The problem is that in many cases there are no more data available to be added. In this case, sometimes it is possible to artificially generate new data. This highly depends on task definition and the format of input data.

In this experiment, the training data used in the previous sections will be artificially extended in order to improve the forecasting performance of the trained model. New

training data will be generated by a slight modification of the original data. The main principle is to add slight randomly generated noise signal to the original time series and merge the result with original training set. In many cases, this action may help the network to learn the dependencies in more general way and prevent the network from overfitting. The more detailed information about this technique, has been described in the following book [9].

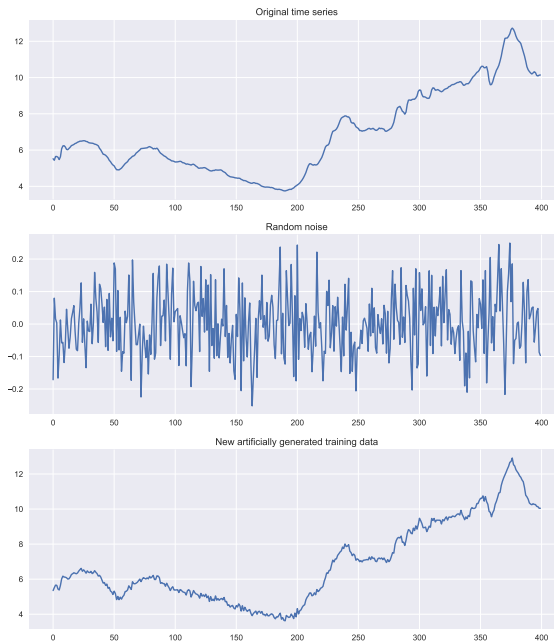


FIGURE 5.12: Example of artificially generated time series data

The ANN model, that has been selected in the previous section has been retrained after the artificial extension of the training set. The performance has been only slightly increased, now the MAPE error is equal to 2.85% for 1-hour forecast horizon, 5.33 for 2-hour forecast horizon and 6.92 for 3-hour forecast horizon. In this specific case, the improvement is not so significant. But nevertheless, it is still an improvement, and sometimes, if it comes to financial issues, even a small improvement may be important.

## 5.7 Experiments summary

The main goal of the experiments in this chapter was to perform the analysis of temperature data and find the optimal forecasting model, that can be further integrated into the other applications. Three different forecasting methods have been used for this task.

At the beginning, analysis of the time series has been performed. It demonstrated the obvious signs of non-stationarity with the clear evidence of trend and seasonality components. The non-stationarity has been confirmed by statistical ADF test. Afterwards, the following preprocessing steps have been required in order to stationarize the series:

- one non-seasonal differencing
- one seasonal differencing with 24-hours lag
- Hampel filter with window size  $k = 7$
- data scaling to  $(-1; 1)$  range (required only for ANN experiments)

The first methods, that have been applied, belong to the class of Autoregressive-Moving-average methods. Performed data analysis helped to make initial suggestion for the proper parameters of the model. In order to select the optimal model, the RMSE measures of individual models have been compared. In the first step, the  $SARIMA(9, 1, 5)x(0, 1, 1)$  model without external factors has been developed. Afterwards, in order to improve the forecast performance the model has been extended by adding the external factors.

The second method was double exponential smoothing, also known as Holt-Winters method. The forecasting method and all additional functions have been completely developed in MATLAB programming environment. The proper parameters of the model have been experimentally selected. The model provides only the ability of time series forecasting without external factors.

The last methods, that have been applied, belong to the class of ANN methods. For the purposes of experiments, the LSTM networks with multiple hidden layers have been used. At first, the model without external factor has been developed. Afterwards, the model has been extended by adding the external factor.

Results demonstrated that forecasting models with external factors are able to make more accurate predictions of local outdoor temperature, that could be intuitively expected. Especially the ANN demonstrated remarkable results. This corresponds to their ability to find out more complex non-linear dependencies in data.

Generally, forecasting of the meteorological indicators belongs to the hardest predictable time series. It is known, that this kind of series can be adequately predicted by the mathematical models only for short forecast horizons. Forecasting for longer periods is pointless. In this cases, there are used more physically oriented techniques at the meteorological stations.



The models without external factors can find-out only the mathematical dependencies between historical and future values of the local outdoor temperature, while the main success of models with external factor is based in their additional ability to find-out the proper correlations between local outdoor temperature and meteorological forecasts. In other words, the models with external factors, in some way adjust the meteorological forecast in such way, that they take into account some local factors, which couldn't be reflected in meteorological forecast.

The ANN forecasting model with the external factor presented the best results, with the MAPE rate equal to 2.91%. The MAPE rate is very popular and informative criterion, when it is necessary to compare the performance of different models and select the better one, but in general, it doesn't express information, whether the model is good or not. The MAPE rate that is good for one series, it doesn't have to be good for other series. In order to decide, whether the model is good or not, the errors should be analyzed a little bit differently.

If the error's plot represents a stationary time series with the constant zero mean, then the model can be considered to be qualitative, and its further improvement is not necessary.

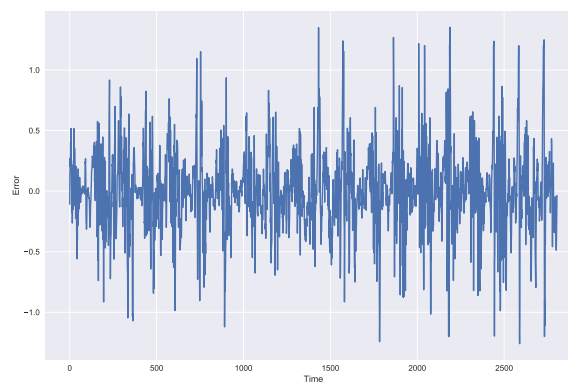


FIGURE 5.13: The plot of forecasting errors calculated on the test data set by the ANN forecasting model with the external factor.

Figure 5.13 demonstrates the plot errors calculated on the test data set by the ANN forecasting model with the external factor. The plot seems to represent the stationary series. This is also confirmed by the plot of ACF for the error vector (Figure 5.14). The ACF drops to zero relatively quickly.

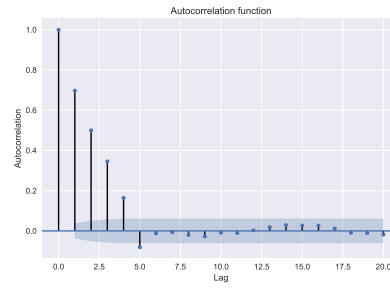


FIGURE 5.14: The plot of forecasting errors calculated on the test data set by the ANN forecasting model with the external factor.

## Chapter 6

# Conclusion

The main issues of this thesis have been to perform the analysis of provided data and to develop the qualitative forecasting models for them. In order to solve this, the theoretical part of the thesis has been devoted to the survey of the time series problematic, forecasting methods, data preprocessing and other important aspects of time series analysis. It was investigated, that very often, the proper data preprocessing plays the key role of the whole process.

In the practical part of the thesis, there have been selected three perspective forecasting methods. Their effectiveness has been demonstratively tested on the internet traffic data set, that is publicly available in the internet. Individual forecasting methods, as well as the time series analysis methods, have proven themselves in the experiments, by demonstration of the remarkable results. After that, these methods could be confidently used for solving the main task of the thesis.

The main task of the thesis is related to the forecasting of the local outdoor temperature of individual buildings. Development of the qualitative forecasting models for the temperature values represents a real practical task and plays an important role for their further integration into the other applications. Forecasting performance of the so called “basic forecasting method”, that simply adjusts the meteorological forecast and is currently used, has served as the benchmark values for the newly developed models. Initially, for the temperature forecasting, there have been used traditional forecasting methods without adding external factors. Afterwards, the models, that make it possible, have been extended by adding an external factor. The meteorological 3-hour forecast values have served as the external factor. All forecasting methods demonstrated relatively good results, and better than the referenced benchmark value. But the models on the base of ANN significantly outperformed any other models, especially the ANN model

with taking into account the external factor. Therefore, they are the most recommended models for the future integration into the other applications.

After the ANN models are already trained, the API of Keras library allows to store the weights and other parameters of network into the text file, and restore it later, in the target application. With a little bit more effort, the forecasting model can be restored to some other ANN libraries.

If the accuracy of the forecast is not such an important criterion, then the other methods can be safely used. For example, all double exponential smoothing models are defined only by two parameters, what makes it easy to integrate them, but nevertheless they still demonstrated relatively good forecasting performance, what could be observed in the experiments.

## 6.1 Discussion about further improvements

One of the possible and the most intuitive improvement is related to the idea of combining together several forecasting models. This technique is usually denoted as “Consensus forecast”, and it is known for being used in fields like econometrics or meteorology. Generally, combining of the forecasts isn’t quite new technique, and for example taking the mean average of the forecasts from different sources, in order to improve the confidence, is being used for a long time. Today, some more sophisticated techniques are being used. The general idea, is to find a linear combination of forecasted values from different models such, that the overall forecasting error will be minimal. For this purpose, it is necessary to construct a proper optimization problem and solve it. The linear programming method can be suitably used for these tasks. The following publication describes the given problematic in more detail [19].

Another option, how to improve the forecasting performance is based on the absolutely different approach. The main idea is, that the training data may contain some sequences, that don’t represent a typical time series behavior, and their presence in the training set contributes to the incorrect estimation of the model, what results into decreasing of the final forecasting performance. This problematic is know as the “anomaly detection” and the following publication may be studied for more details [20]. The general idea is to remove the anomalies from the training data and make them more reliable.

# Bibliography

- [1] Gregory C. Reinsel Greta M. Ljung George E. P. Box, Gwilym M. Jenkins. *Time Series Analysis: Forecasting and Control*. Wiley, fifth edition, aug 2015.
- [2] Michael Falk. *A First Course on Time Series Analysis — Examples with SAS*. Chair of Statistics, University of Wurzburg, aug 2012.
- [3] James J. Filliben. Autocorrelation. URL <http://www.itl.nist.gov/div898/handbook/eda/section3/eda35c.htm>.
- [4] Brian Borchers. The partial autocorrelation function, april 2001. URL <http://www.ees.nmt.edu/outside/courses/GEOP505/Docs/pac.pdf>.
- [5] Jaakko Astola Ronald K. Pearson, Yrjo Neuvo and Moncef Gabbouj. Generalized hampel filters, may 2017. URL [https://tutcris.tut.fi/portal/files/7991765/Generalized\\_Hampel\\_Filters.pdf](https://tutcris.tut.fi/portal/files/7991765/Generalized_Hampel_Filters.pdf).
- [6] Simon S. HAYKIN. *Neural networks and learning machines*. New York: Prentice Hall,, third edition, 2009.
- [7] University of Maryland prof. Charles Stangor. The neuron is the building block of the nervous system, may 2017. URL <http://2012books.lardbucket.org/books/beginning-psychology/s07-01-the-neuron-is-the-building-blo.html>.
- [8] Commonly used activation functions, may 2017. URL <http://cs231n.github.io/neural-networks-1/>.
- [9] Michael A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015.
- [10] Tie Liu. Application of markov chains to analyze and predict the time series, may 2017. URL [http://www.ccsenet.org/journal/index.php/mas/article/viewFile/6040/4874\\_1\\_1](http://www.ccsenet.org/journal/index.php/mas/article/viewFile/6040/4874_1_1).
- [11] Wang Jiahui Zivot Eric. *Modeling financial time series with S-plus. 2nd ed. New York*. Springer, 2006.

- 
- [12] Robert Nau. Statistical forecasting: notes on regression and time series analysis, may 2017. URL <http://people.duke.edu/~rnau/411home.htm>.
- [13] Eduardo Rossi. Unit roots, may 2017. URL [http://economia.unipv.it/pagp/pagine\\_personali/erossi/rossi\\_unit\\_roots\\_PhD.pdf](http://economia.unipv.it/pagp/pagine_personali/erossi/rossi_unit_roots_PhD.pdf).
- [14] Robert Nau. The logarithm transformation, may 2017. URL <http://people.duke.edu/~rnau/411log.htm>.
- [15] Robert Nau. Seasonal differencing in arima models, may 2017. URL <https://people.duke.edu/~rnau/411sdif.htm>.
- [16] Robert Nau. Summary of rules for identifying arima models, may 2017. URL <http://people.duke.edu/~rnau/arimrule.htm>.
- [17] Jason Brownlee. Time series prediction with lstm recurrent neural networks in python with keras, july 2016. URL <http://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/>.
- [18] Petros Kritharas. Developing a sarimax model for monthly wind speed forecasting in the uk, 2013. URL <https://dspace.lboro.ac.uk/dspace-jspui/bitstream/2134/16350/2/Thesis-2014-Kritharas.pdf>.
- [19] MSc Apostolos Panagiotopoulos. Optimising time series forecasts through linear programming, december 2011. URL [http://eprints.nottingham.ac.uk/12515/1/Apostolos\\_Panagiotopoulos\\_Thesis.pdf](http://eprints.nottingham.ac.uk/12515/1/Apostolos_Panagiotopoulos_Thesis.pdf).
- [20] Deepthi Cheboli. Anomaly detection of time series, may 2010. URL <http://conservancy.umn.edu/bitstream/handle/11299/92985/?sequence=1>.