ulm university universität
uulm

Faculty of Mathematics and Economics
Institute of Mathematical Finance

Time series models for predicting failure-relevant characteristics of railway switches

Master's thesis in Mathematics at the University of Ulm

Submitted by:
Daniel Fässler
746472

Referees:
Prof. Dr. Alexander Lindner
Dr. Thorsten Neumann

Advisor:
Dr. Daniela Narezo Guzmán

July 2020

Acknowledgements

I would like to express my sincere gratitude to my two advisors Dr. Daniela Narezo Guzmán and Dr. Thorsten Neumann from the German Aerospace Center's Institute of Transportation Systems. I was able to learn a great deal through their continuous support and motivation, amongst other things about railway switches and railway structure in general.
Furthermore, I want to thank Prof. Dr. Alexander Lindner from the University of Ulm for helping me with topics in time series analysis, advising me and giving me the opportunity to write this thesis.
This work would not have been possible without the support of the European joint undertaking Shift2Rail in the context of the projects In2Smart (Grant No. 730569) and In2Smart2 (Grant No. 881574). I want to thank the Dutch company Strukton Rail, namely Edin Hadzic and Henk Samson for providing the POSS measurement data for this study.
Lastly, I want to thank my family and friends who supported me on the long road and specifically with regard to this thesis Mark Bangert, who helped me with corrections.

Abstract

Railway switches are a crucial part of the railway infrastructure whose proper functioning is essential for safe and trouble-free rail traffic. The current consumption of a point motor responsible for a switch movement plays an important role in anomaly detection and predictive maintenance of a railway switch. The power consumed by point engines of railway switches depends on different factors, such as air temperature or humidity. This thesis presents various time series models that incorporate exogenous variables to model the total power consumed by railway switches. Those models enable the recognition of abnormal behaviour by railway switches to prevent failures through timely initiation of maintenance actions.

Contents

List of Figures

List of Tables

1 Introduction, motivation and aim of the thesis

Delays in rail transport cause enormous economic damage of a hardly quantifiable amount and simultaneously unpunctuality is one of the biggest annoyances for passengers [1, p. 1]. Reducing maintenance costs and increasing punctuality have been important to the railway infrastructure managers of the European Union for many years and continue to be so [1, p. 1]. One of the main causes of infrastructural disruptions of rail transports are railway switches. Railway switches are central elements of the railway infrastructure (c.f. Figure 1.1) whose faultless functioning is essential for smooth and safe rail transport. At the same time, they are susceptible to faults and defects due to their complex construction with high mechanical loads and environmental influences (c.f. [12]).

Figure 1.1: An example of a railway switch (left) and in enlarged view (right)

The negative effects of those infrastructural disruptions can be prevented if a malfunction of the switch can be diagnosed in time and effective as well as efficient

countermeasures can be planned and executed [1, p. xix]. Predictive maintenance of the corresponding elements of the switch shall lead to asset life extension, cost reduction and an increasement of the availability of railway transportation in general [16, p. 1]. In practice, several systems for switch monitoring including the Sidis W system from Siemens [2] and the measurement system POSS [28] of the Dutch construction and rail company Strukton Rail have been in existence for some time. The goal is reliable automated condition monitoring, anomaly detection ([15, 16]) as well as a (model-based) Fault Cause Diagnosis ([23, 24]). The principle behind it is to continuously measure (e.g. with 50Hz) the current of an electrical motor that is responsible for moving the switch blades from one position to the other. This correlates with the force that is needed to adjust the switch tongues. The resulting current curves (c.f. Figure 1.2) may vary but can be divided into various sections. The highest power consumption is usually at the beginning, i.e. when starting the engine and releasing the switch lock. During the actual movement of the tongues, the current consumption remains nearly constant at a medium level. At the end of the switch movement, the current decreases and when the switch tongues get locked back the motor shuts down and the measured current returns to zero amperes.

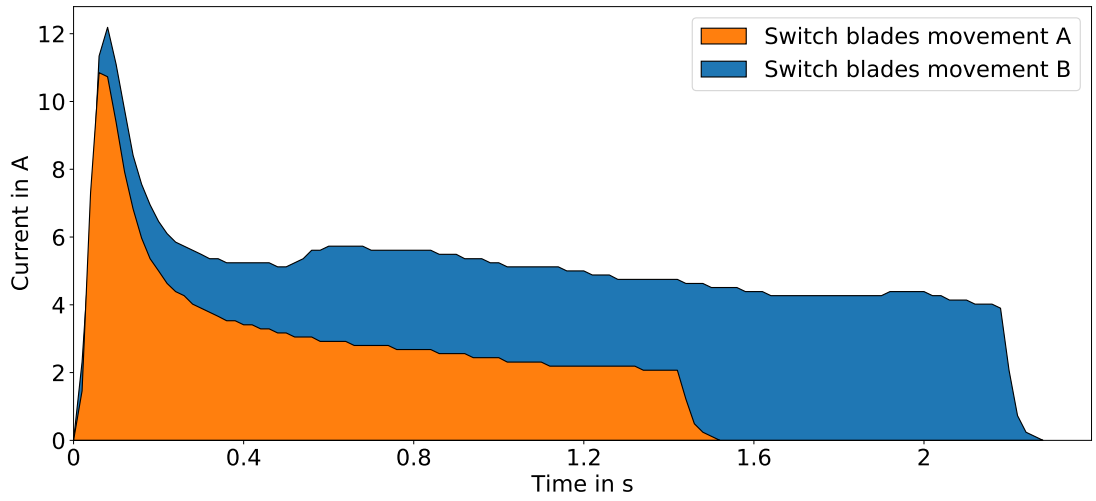

Figure 1.2: Two discrete current curves of different switch blade movements

Conspicuous features of the current curve can be that the measured current during the actual movement of the tongues is higher than expected or the duration of the

switch movement is longer than usual (c.f. Figure 1.2 Switch blades movement B). In addition to more complex approaches to anomaly detection (c.f. [23, 24]), in many cases the integral of the current curve, i.e. the area below the curve (c.f. Figure 1.2) can be used as an indicator of possible faults [10]. With regard to the prediction of switch anomalies and failures, it is also appropriate if not even necessary to monitor the temporal course of the integral values or comparable from derived feature parameters of the current curve, which is what this work focuses on. This area under the current curve can be seen as the total power consumed by the electrical motor of the switch. When only finite measurements of the current of the electric motor are available, an approximative value of the total power can be used (e.g. the sum of current values as a discrete approximation) as a derived feature parameter. The goal of this thesis is to model and analyse the timely progress of total power (or its discrete version) mathematically over time and then predict future model-based values. The predicted values can then be used for failure diagnosis, for example with pre-defined thresholds. Deviations between previously predicted normal behaviour and actual values can also be used for fault detection to indicate gradually developing defects (degradation effects) to prevent a failure of the switch by a timely performed maintenance action. Well studied and applied models in classical time series analysis are Autoregressive Moving Average (ARMA) models. For instance, ARMA models have been used by several researchers for short-term forecasting of electrical loads (c.f. [17, 20], [22, p. 2]). They have also been used for predicting electricity demand loads [26], predicting household electricity consumption [6] and in terms of predicting failures of railway switches García et al. used a modified ARMA model (VARMA) to forecast the duration of a switch movement [14]. The wide class of ARMA models and some extensions will be defined and discussed in view of the current consumption of railway switches in this thesis.

2 Linear models for describing a time series

2.1 ARMA models

In this chapter, we want to introduce ARMA processes and some extensions which are widely used and form a useful class of models to describe the dynamics of an individual time series. We will always assume that the time parameter T is equal to $\mathbb{N}_0$ or $\mathbb{Z}$.

2.1.1 Stationarity and properties of an ARMA process

Definition 1. *Let $X = (X_t)_{t\in T}$ be a time series with $\mathbb{E}\, X_t^2 < \infty$ for all $t \in T$. Then*

$$\mu_X(t) := \mathbb{E}X_t, \quad t \in T$$

is called the mean function and

$$\gamma_X(r,s) := \mathrm{Cov}(X_r, X_s) = \mathbb{E}[(X_r - \mu_X(r))(X_s - \mu_X(s))], \quad r,s \in T,$$

the covariance function of X.

Definition 2. *A real-valued time series $X = (X_t)_{t\in\mathbb{Z}}$ is said to be weakly stationary if*

(i) $\mathbb{E}X_t^2 < \infty$ *for all* $t \in \mathbb{Z}$,

(ii) $\mathbb{E}X_t = \mathbb{E}\, X_{t'}$ *for all* $t, t' \in \mathbb{Z}$,

(iii) $\gamma_X(t+h,t) = \gamma_X(t'+h,t')$ *for all* $t, t', h \in \mathbb{Z}$*, i.e.* $\gamma_X(r,s)$ *only depends on* $r-s$*.*

Then $\mu_X := \mathbb{E}(X_t)$ *is called the mean of* X*,*

$$\gamma_X(h) := \gamma_X(h,0) = \mathrm{Cov}(X_{t+h}, X_t) \quad \textit{for all } t, h \in \mathbb{Z}$$

is called the autocovariance of X *at lag* h *and* $\gamma_X(.)$ *is called the autocovariance function of* X *(ACVF). The autocorrelation function* $\rho_X(.)$ *of the stationary time series* X *(ACF) is defined analogously (if* $\gamma_X(0) \neq 0$*) as the function whose value at lag* h *is*

$$\rho_X(h) := \frac{\gamma_X(h)}{\gamma_X(0)} = \mathrm{Cor}(X_{t+h}, X_t) \quad \textit{for all } t, h \in \mathbb{Z}.$$

Remark 1. *Stationarity defined as above is frequently referred to as covariance stationarity, stationarity in the wide case, or second-order stationarity. However, there are other stronger definitions of stationarity such as strict stationarity. We sometimes call a weakly stationary time series simply stationary in this thesis.*

Example 1 (White Noise)**.** *Let* $\mathbb{E}\, X_t^2 < \infty$, $\mathbb{E}\, X_t = 0$, $\mathrm{Var}(X_t) = \mathbb{E}X_t^2 := \sigma^2$, $\sigma^2 \in (0, \infty)$ *for all* $t \in T$ *and* $\mathrm{Cov}(X_t, X_t') = 0$ *for all* $t \neq t'$*, then* $(X_t)_{t\in T}$ *is called white noise, written as*

$$(X_t)_{t\in T} \sim \ \mathrm{WN}(0, \sigma^2).$$

This time series is weakly stationary with covariance function

$$\gamma_X(h) = \begin{cases} \sigma^2, & h = 0, \\ 0, & h \neq 0. \end{cases}$$

Definition 3 (ARMA process)**.** *Let* $p, q \in \mathbb{N}_0$.

(i) A time series $(X_t)_{t\in\mathbb{Z}}$ *is called ARMA*(p,q) *process if* $(X_t)_{t\in\mathbb{Z}}$ *is weakly stationary and coefficients* $\phi_1, \ldots, \phi_p \in \mathbb{C}$, $\theta_1, \ldots, \theta_q \in \mathbb{C}$ *as well as a complex-valued white noise* $(Z_t)_{t\in\mathbb{Z}}$ *exist such that*

$$X_t = \phi_1 X_{t-1} + \ldots + \phi_p X_{t-p} + Z_t + \theta_1 Z_{t-1} + \ldots + \theta_q Z_{t-q}, \quad t \in T. \tag{2.1}$$

(ii) The coefficients $\phi_1, \ldots, \phi_p$ *are called autoregressive coefficients and* $\theta_1, \ldots, \theta_q$ *moving average coefficients of (2.1).*

It can be advantageous to write an ARMA process in a more compact form.

Definition 4. *Let ϕ and θ be the pth and qth degree polynomials*

$$\phi(z) := 1 - \phi_1 z - \ldots - \phi_p z^p, \quad z \in \mathbb{C}$$

and

$$\theta(z) := 1 + \theta_1 z + \ldots + \theta_q z^q, \quad z \in \mathbb{C}.$$

The equation (2.1) can be written in a more compact form as

$$\phi(B)X_t = \theta(B)Z_t, \quad t \in \mathbb{Z}, \tag{2.2}$$

where B is the backshift operator, defined by

$$B^j X_t := X_{t-j}.$$

The polynomials ϕ and θ are called the autoregressive and moving average polynomials respectively of the equation (2.2). Both are called characteristic polynomials. The equation (2.2) is called ARMA equation.

Theorem 1. *Let $\phi(z)$ and $\theta(z)$ be the autoregressive and moving average polynomials of the ARMA equation (2.2) and let $\phi(z) \neq 0$ for all $z \in \mathbb{C}$ with $|z| = 1$. Furthermore let $(Z_t)_{t\in\mathbb{Z}} \sim \mathrm{WN}(0, \sigma^2)$. Then a unique stationary solution $(X_t)_{t\in\mathbb{Z}}$ exists for the equation*

$$\phi(B)X_t = \theta(B)Z_t, \quad t \in \mathbb{Z}.$$

An ARMA(p, q) process therefore exists with these characteristic polynomials. The solution is given by

$$X_t = \sum_{j=-\infty}^{\infty} \psi_j Z_{t-j}, \quad t \in \mathbb{Z},$$

where the coefficients $(\psi_j)_{j\in\mathbb{Z}}$ are the coefficients of the absolutely convergent Laurent series

$$\psi(z) = \sum_{j=-\infty}^{\infty} \psi_j z^j = \frac{\theta(z)}{\phi(z)},$$

on a specified annulus $\{\delta < |z| < \delta^{-1}\}, \quad \delta > 0$.

Proof. See [4, Theorem 3.1.3]. □

Remark 2. *One can show with spectral analysis (c.f. [4, Problem 4.28.]), that if* $\phi(.)$ *and* $\theta(.)$ *are polynomials with no common zeros and if* $\phi(z) = 0$ *for some* $z \in \mathbb{C}$ *such that* $|z| = 1$*, the ARMA equation (2.2) has no stationary solution. If* $(X_t)_{t\in\mathbb{Z}}$ *is an ARMA*(p, q) *process for which the polynomials* $\phi(.)$ *and* $\theta(.)$ *have common zeros, then there are two possibilities (c.f. [4, p. 86]):*

1. *None of the common zeros lie on the unit circle and a unique weakly stationary solution of the ARMA equation exists.*
2. *At least one of the common zeros lies on the unit circle and the ARMA equation may have more than one weakly stationary solution or none.*

Example 2 (MA(q) process)**.** *Let* $X = (X_t)_{t\in T}$ *and* $\phi(z) \equiv 1$*. Then*

$$X_t = \theta(B)Z_t$$

is said to be a moving average process of order q*. It holds that* $\mathbb{E}X_t^2 < \infty$ *and* $\mathbb{E}X_t = 0$*. Clearly* $\phi(z) \neq 0$ *for all* $z \in \mathbb{C}$ *and thus a moving average process is weakly stationary according to Theorem 1. The autocovariance function is given by (defining* $\theta_0 := 1$ *and* $\theta_j = 0$ *for* $j > q$*):*

$$\gamma_X(h) = \begin{cases} \sigma^2 \sum_{j=0}^{q-|h|} \theta_j \theta_{|h|+j} & \text{if } |h| \leq q, \\ 0 & \text{if } |h| > q. \end{cases}$$

Example 3 (AR(p) process)**.** *Let* $X = (X_t)_{t\in T}$ *and* $\theta(z) \equiv 1$*. Then*

$$\phi(B)X_t = Z_t$$

is said to be an autoregressive process of order p*. In this case, the existence and uniqueness of a stationary solution needs closer investigation. We illustrate it by examing the case where* $\phi(z) = 1 - \phi_1 z$ *(AR(1) process), i.e.*

$$X_t = Z_t + \phi_1 X_{t-1}.$$

According to Theorem 1 and Remark 2, a stationary solution exists if $|\phi_1| > 1$ *or* $|\phi_1| < 1$*. No stationary solution exists for* $|\phi_1| = 1$*.*

2.1.2 The ACF and PACF of an ARMA process

We want to discuss useful tools to determine the parameters of an ARMA process.

Definition 5 (Causal ARMA process)**.** *An ARMA*(p, q) *process* $(X_t)_{t\in\mathbb{Z}}$*, defined by* $\phi(B)X_t = \theta(B)Z_t,\ t \in \mathbb{Z}$ *is said to be causal with respect to the white noise* $(Z_t)_{t\in\mathbb{Z}}$ *if a sequence of constants* $(\psi_j)_{j\in\mathbb{N}_0}$ *exists such that* $\sum_{j=0}^{\infty} |\psi_j| < \infty$ *and*

$$X_t = \sum_{j=0}^{\infty} \psi_j Z_{t-j}, \quad t \in \mathbb{Z}. \tag{2.3}$$

That means in case of causality of an ARMA(p, q) process, $(X_t)_{t\in\mathbb{Z}}$ only depends on the present and past.

Theorem 2. *Let* $X = (X_t)_{t\in\mathbb{Z}}$ *be an ARMA*(p, q) *process for which the polynomials* $\phi(.)$ *and* $\theta(.)$ *have no common zeros. Then* X *is causal if and only if* $\phi(z) \neq 0$ *for all* $z \in \mathbb{C} : |z| \leq 1$*. The coefficients of the sequence* $(\psi_j)_{j\in\mathbb{N}_0}$ *of (2.3) are determined by the relation*

$$\psi(z) = \sum_{j=0}^{\infty} \psi_j z^j = \frac{\theta(z)}{\phi(z)}, \quad |z| \leq 1.$$

Proof. See [4, Theorem 3.1.1]. □

Example 4. *The AR(1) process of Example 3 is causal if and only if* $|\phi_1| < 1$ *according to Theorem 2.*

In case of causality, the autocovariance function of the real-valued ARMA(p, q) process of Definition 5 can be directly calculated and is given by (c.f. [4, p. 91]):

$$\gamma_X(h) = \sigma^2 \sum_{j=0}^{\infty} \psi_j \psi_{j+|h|}.$$

Therefore the autocorrelation function is given by:

$$\rho_X(h) = \frac{\sum_{j=0}^{\infty} \psi_j \psi_{j+|h|}}{\sum_{j=0}^{\infty} \psi_j^2}.$$

If it is a pure MA(q) process it holds that $\psi_0 = 1$, $\psi_j = \theta_j$ for $j = 1, \ldots, q$ and $\psi_j = 0$ for $j > q$. Therefore, for a pure MA(q) process the ACF $\rho_X(h) = 0$ for $|h| > q$, so that we can use the ACF for determining the order of q (c.f. [8, p.18]).
Given an AR(p) process the ACF is not helpful to determine the order of p because it does not zero-out as in the MA(q) case. A useful measure to determine the order of an AR(p) process is the partial autocorrelation function (PACF).

Definition 6. *The partial autocorrelation function $\alpha : \mathbb{N} \to \mathbb{R}$ of a real-valued weakly stationary time series $(X_t)_{t\in T}$ is defined by*

$$\alpha(k) = \begin{cases} \rho_X(1) = \mathrm{Cor}(X_{t+1}, X_t) & \text{for } k = 1, \\ \mathrm{Cor}(X_{t+k} - P_{k-1}X_{t+k}, X_t - P_{k-1}X_t) & \text{for } k \geq 2, \end{cases}$$

where P_{k-1} denotes the projection onto $P_{\overline{sp}\{1, X_{t+1}, \ldots, X_{t+k-1}\}}$ which is the closed span of the set $\{1, X_{t+1}, \ldots, X_{t+k-1}\}$.

The PACF is seen as the correlation between X_{t+k} and X_t, where the information of $X_{t+2}, \ldots, X_{t+k-1}$ is considered. For more information and how to find the projections $P_{k-1}X_{t+k}$ and $P_{k-1}X_t$, we refer to [8, Example 1.32] and [4, §2.7].

Example 5. *Let $X = (X_t)_{t\in T}$ be a zero-mean $AR(1)$ process:*

$$X_t = \phi X_{t-1} + Z_t.$$

We have $\alpha(1) = \phi$ and

$$\begin{aligned} \alpha(2) &= \mathrm{Cor}(X_{t+2} - P_1X_{t+2}, X_t - P_1X_t) = \mathrm{Cor}(X_{t+2} - \phi X_{t+1}, X_t - P_1X_t) \\ &= \mathrm{Cor}(Z_{t+2}, X_t - P_1X_t) = 0 \end{aligned}$$

because $X_t - P_1X_t$ is a function of $Z_t, Z_{t-1}, \ldots$ and they are uncorrelated with Z_{t+2}. Similar calculations show that $\alpha(k) = 0$ for $k > 1$.

For a general AR(p) process, it can be shown by similar calculations as in Example 5, that $\alpha(k) = 0$ for $k > p$. Estimates exist for the ACF and PACF. In practice, we can plot the sample ACF and sample PACF of our data to see if they might be samples of an AR(p) or MA(q) process. At the top of Figure 2.1 are 1000 samples of an AR(1) and MA(1) process visualised as well as their sample ACF and PACF. We can see that the sample partial autocorrelation function lies in or near the blue 5% confidence interval for $k > 1$ whereas the sample autocorrelation function slowly decays in case of the AR(1) process. The same applies in reverse for the simulated MA(1) process.

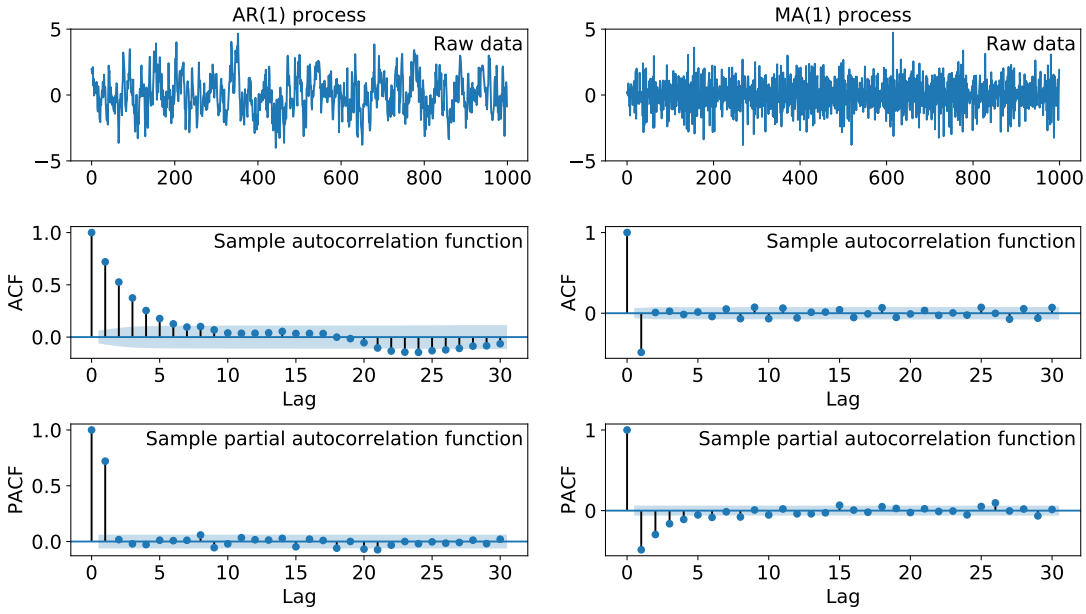

Figure 2.1: A simulated AR(1) process and the corresponding sample ACF and PACF (left) and in an analogous way a MA(1) process (right). Both with coefficient -0.7.

2.2 Extensions and covariates of ARMA models

2.2.1 ARIMA processes

When we introduced ARMA processes, we assumed that they are weakly stationary. This definition can be generalised to a wider class which incorporates nonstationary

time series which is provided by integrated ARMA processes, i.e. Autoregressive Integrated Moving Average processes which can be reduced to ARMA processes when differenced finitely many times.

Definition 7. *Let $d \in \mathbb{N}_0$. The time series $(X_t)_{t\in T}$ is called an ARIMA(p, d, q) process if $\nabla^d X_t := (1 - B)^d X_t$ is a causal ARMA(p, q) process (if $d = 0$ the ARIMA($p, 0, q$) process is defined as an ARMA(p, q) process).*

2.2.2 ARMA models with exogenous variables

Sometimes it can be beneficial to use a model that includes exogenous variables.

Definition 8. *The time series $(X_t)_{t\in T}$ is called ARMAX(p, q, b) process if it has the form of (2.1), i.e. it is weakly stationary and contains the AR(p) and MA(q) terms and additionally a linear combination of the last b inputs of an external time series $(U_t)_{t\in T}$:*

$$X_t = \sum_{i=1}^{p} \phi_i X_{t-i} + Z_t + \sum_{j=1}^{q} \theta_j Z_{t-j} + \sum_{k=1}^{b} \eta_k U_{t-k}, \quad t \in T, \tag{2.4}$$

where $\eta_1, \ldots, \eta_b \in \mathbb{C}$ are the coefficients of the external time series $(U_t)_{t\in T}$.

2.2.3 Vector-valued models

If we want to forecast a $(k \times 1)$ vector-valued time series $X_t = (X_{t1}, \cdots, X_{tk})'$, we can extend the ARMA model to the multivariate case (c.f. [8, p. 25]).

Definition 9. *A $(k \times 1)$ vector-valued time series $(X_t)_{t\in T}$ is called a VARMA(p, q) process if it is stationary and*

$$X_t = \nu + A_1 X_{t-1} + \ldots + A_p X_{t-p} + Z_t + B_1 Z_{t-1} + \ldots + B_q Z_{t-q}, \quad t \in T, \tag{2.5}$$

where ν is a $k-$dimensional constant, $A_1, \ldots, A_p$, $B_1, \ldots, B_q$ are $(k \times k)$ matrices, $A_p \neq 0$, $B_q \neq 0$ and $(Z_t)_{t\in T}$ is a vector-valued white noise process which is multivariate normally distributed with mean-zero and non-singular covariance matrix Σ_z. The model is called a VAR(p) process if If $q = 0$ and a VMA(q) process if $p = 0$.

We can also add a combination of matrices and exogenous vectors to the VARMA model to get a vector-valued model with exogenous inputs.

Definition 10. *A $(k \times 1)$ vector-valued time series $(X_t)_{t\in T}$ is called VARMAX(p, q, b) process if it has the form of (2.5), i.e. it is weakly stationary and contains the VAR(p) and VMA(q) terms and additionally a combination of actual and the last b inputs of an external $k-$dimensional time series $(U_t)_{t\in T}$:*

$$X_t = \nu + \sum_{i=1}^{p} A_i X_{t-i} + Z_t + \sum_{j=1}^{q} B_j Z_{t-j} + \sum_{k=0}^{b} M_k U_{t-k}, \quad t \in T, \tag{2.6}$$

where ν is a k-dimensional constant, $A_1, \ldots, A_p$, $B_1, \ldots, B_q$ are $(k \times k)$ matrices and $M_0, \ldots, M_b$ are $(r \times r)$ matrices with $r \leq k$. The matrices $A_p,\ B_q,\ U_b$ are non-zero and $(Z_t)_{t\in T}$ is a vector-valued white noise process which is multivariate normally distributed with mean-zero and non-singular covariance matrix Σ_z. The model is called a VARX(p) model if $q = 0$ and a VMAX(q) model if $p = 0$.

2.2.4 Seasonal ARIMA models

Some time series are characterised by a serial correlation at a seasonal lag [4, pp. 320-326]. We define it as in [4, p. 323].

Definition 11. *Let c and D be non-negative integers, then $(X_t)_{t\in T}$ is said to be a SARIMA(p, d, q)$\times$$(P, D, Q)_s$ process (seasonal ARIMA) with period s if the differenced process $Y_t := (1 - B)^d(1 - B^s)^D X_t$ is a causal ARMA process,*

$$\phi(B)\Phi(B^s)Y_t = \theta(B)\Theta(B^s)Z_t, \quad Z_t \sim \mathrm{WN}(0, \sigma^2),$$

where $\phi(z) = 1 - \phi_1 z - \ldots - \phi_p z^p,\ \Phi(z) = 1 - \Phi_1 z - \ldots - \Phi_p z^p$,
$\theta(z) = 1 + \theta_1 z + \ldots + \theta_z z^q$ and $\Theta(z) = 1 + \Theta_1 z + \ldots + \Theta_Q z^Q$.

2.2.5 Linear regression with ARMA errors

Definition 12. *Consider the usual regression model for a univariate time series $(Y_t)_{t\in T}$, given by*

$$Y_t = X_t\beta + \xi_t, \quad t = 1, \ldots, n, \tag{2.7}$$

where X_t is a $(1 \times k)$ regression vector of exogenous variables, β a $(k \times 1)$ vector of regression coefficients and ξ_t is an error term. We call this model a linear regression model with ARMA errors if the error term follows an ARMA model of form (2.1).

2.3 Methods to check an autoregressive model for stationarity

Many methods to check an autoregressive model for stationarity are derived from unit root tests such as the Dickey-Fuller test or the Phillips-Perron test. We want to introduce the Dickey-Fuller test based on [25, 11]. For more information, we refer to [25, pp. 119-131] and [11, pp. 221-224]. Both tests are based on the regression of X_t on X_{t-1}, whereby in some circumstances deterministic variables like a constant or a linear trend are used additionally. Let us first consider a real-valued AR(1) process:

$$X_t = \phi_1 X_{t-1} + Z_t,$$

or equivalently the Dickey-Fuller regression:

$$\Delta X_t := \beta X_{t-1} + Z_t,$$

where $\beta = \phi_1 - 1$. As seen in Example 3 this model is not stationary for $\phi_1 = 1$. For both tests, the null hypothesis consists of the assumption that a unit root exists and thus the model is not stationary. The alternative hypothesis is that the process is stationary. The null hypothesis and alternative hypothesis can be written as:

$$\mathrm{H}_0 : \phi_1 = 1 \qquad \text{vs.} \qquad \mathrm{H}_1 : -1 < \phi_1 < 1$$

or in case of the Dickey-Fuller regression:

$$\mathrm{H}_0 : \beta = \phi_1 - 1 = 0 \qquad \text{vs.} \qquad \mathrm{H}_1 : -2 < \beta = \phi_1 - 1 < 0.$$

Dickey and Fuller considered three different regression equations that can be used to test the time series for the presence of a unit root (c.f. [7]):

$$\Delta X_t = \phi_1 X_{t-1} + Z_t, \tag{2.8}$$

$$\Delta X_t = \alpha + \phi_1 X_{t-1} + Z_t, \tag{2.9}$$

$$\Delta X_t = \alpha + \delta t + \phi_1 X_{t-1} + Z_t. \tag{2.10}$$

In the first case (2.8, random walk), where the Dickey-Fuller regression includes no constant, the rejection of the null hypothesis implies that $(X_t)_{t\in T}$ might be a stationary process with zero-mean. This specification is only recommended to be used if it is ensured that the given data has zero-mean. If the given data exhibits no trend, then the Dickey-Fuller test with a constant ((2.9), random walk with drift) should be used and the rejection of the null hypothesis implies that the process might be stationary with mean $\mu = \frac{\alpha}{1-\phi_1}$. Does the data show a trend, then (2.10, random walk with drift and deterministic trend) should be used and if the null hypothesis gets rejected it implies that $(X_t)_{t\in T}$ might be trend-stationary (stationary after eliminating a trend). The test statistic is $n(\hat{\rho} - 1)$, where n is the sample size and $\hat{\rho}$ depends on the form of the regression. This test statistic is not asymptotically normally distributed. However, critical values can be found in tabular form in [13] or [18]. Additionally, three test statistics to test joint hypotheses on the coefficients are provided which are not asymptotically F-distributed but can also be found in [18]. Since many time series show a high autocorrelation, it is necessary to extend the first-order autoregressive process to pth-order autoregressive processes. Delayed differences $\Delta X_{t-1}, \ldots, \Delta X_{t-p+1}$ need to be added to the Dickey-Fuller regression:

$$\Delta X_t = \alpha + \delta t + \phi_1 X_{t-1} + \gamma_1 \Delta X_{t-1} + \ldots + \gamma_{p-1} \Delta X_{t-p+1} + Z_t.$$

The equations (2.8), (2.9) and (2.10) are replaced by the autoregressive processes:

$$\Delta X_t = \phi_1 X_{t-1} + \sum_{i=2}^{p} \gamma_i \Delta X_{t-i+1} + Z_t, \tag{2.11}$$

$$\Delta X_t = \alpha + \phi_1 X_{t-1} + \sum_{i=2}^{p} \gamma_i \Delta X_{t-i+1} + Z_t, \tag{2.12}$$

$$\Delta X_t = \alpha + \delta t \phi_1 X_{t-1} + \sum_{i=2}^{p} \gamma_i \Delta X_{t-i+1} + Z_t. \tag{2.13}$$

With this approach we obtain a test often referred to as the augmented Dickey-Fuller test or ADF test. The limit distributions for the test of the hypothesis do not change through this autoregressive correction. However, for the coefficients of the correction terms the standard F-statistics can now be used. This also applies if the correction terms not only contain autoregressive but also moving average terms.

3 State space models and the Kalman filter

3.1 Forecasting methods and principles

3.1.1 Principles of forecasting

This subsection is important for evaluating a forecast of our time series. We follow the ideas of [18, pp. 72-74], assuming that we are interested in the forecast of a variable X_{t+1}, based on a set of variables Y_t, which we observed at time t. Let $\hat{X}_{t+1|t}$ be the forecast of X_{t+1} based on Y_t (for example the recent n values of X_{t+1}). To evaluate the forecast $\hat{X}_{t+1|t}$ we need a suitable loss function, to specify how sure we are with our prediction. An appropriate approach is given by a quadratic loss function, that means we want to choose the forecast $\hat{X}_{t+1|t}$, such that the Mean Squared Error (MSE) associated with the forecast $\hat{X}_{t+1|t}$ of X_{t+1},

$$\text{MSE}(\hat{X}_{t+1|t}) := \mathbb{E}(X_{t+1} - \hat{X}_{t+1|t})^2$$

is minimal.

Theorem 3. *Let $\hat{X}_{t+1|t}$ be the forecast of X_{t+1} based on a set of variables Y_t with the smallest mean squared error among all forecasts of X_{t+1} based on Y_t. It holds that*

$$\hat{X}_{t+1|t} = \mathbb{E}(X_{t+1}|Y_t).$$

Proof. Let us consider basing $\hat{X}_{t+1|t}$ on any function $g(Y_t)$ other than the conditional expectation,

$$\hat{X}_{t+1|t} = g(Y_t).$$

The MSE of $g(Y_t)$ is given as

$$\begin{aligned}\mathbb{E}[X_{t+1} - g(Y_t)]^2 &= \mathbb{E}[X_{t+1} - \mathbb{E}(X_{t+1}|Y_t) + \mathbb{E}(X_{t+1}|Y_t) - g(Y_t)]^2 \\ &= \mathbb{E}[X_{t+1} - \mathbb{E}(X_{t+1}|Y_t)]^2 \\ &+ 2\,\mathbb{E}\{[X_{t+1} - \mathbb{E}(X_{t+1}|Y_t)][\mathbb{E}(X_{t+1}|Y_t) - g(Y_t)]\} \\ &+ \mathbb{E}\{[E(X_{t+1}|Y_t) - g(Y_t)]^2\}.\end{aligned} \tag{3.1}$$

First, we write the second summand of (3.1) as

$$2\,\mathbb{E}(\eta_{t+1}),$$

where

$$\eta_{t+1} := \{[X_{t+1} - \mathbb{E}(X_{t+1}|Y_t)][\mathbb{E}(X_{t+1}|Y_t) - g(Y_t)]\}$$

and consider the expectation of η_{t+1} conditional on Y_t. The terms $\mathbb{E}(X_{t+1}|Y_t)$ and $g(Y_t)$ are known constants conditional on Y_t and we can factor them out:

$$\mathbb{E}(\eta_{t+1}|Y_t) = [\mathbb{E}(X_{t+1}|Y_t) - g(Y_t)] \cdot \mathbb{E}([X_{t+1} - \mathbb{E}(X_{t+1}|Y_t)]|Y_t) = 0.$$

According to the law of iterated expectation it follows that

$$\mathbb{E}(\eta_{t+1}) = \mathbb{E}_{Y_t}(E[\eta_{t+1}|Y_t]) = 0.$$

That simplifies equation (3.1) to

$$\mathbb{E}[X_{t+1} - g(Y_t)]^2 = \mathbb{E}[X_{t+1} - \mathbb{E}(X_{t+1}|Y_t)]^2 + \mathbb{E}([\mathbb{E}(X_{t+1}|Y_t) - g(Y_t)]^2). \tag{3.2}$$

It follows that the function $g(Y_t)$ that minimises the mean squared error (3.2), is the function that sets the second term in (3.2) to zero. Thus it follows that

$$\mathbb{E}(X_{t+1}|Y_t) = g(Y_t).$$

□

3.1.2 Forecasts based on a linear projection

Now we restrict the set of forecasts by assuming that the forecast $\hat{X}_{t+1|t}$ is a linear function of the n-dimensional vector Y_t:

$$\hat{X}_{t+1|t} = \alpha' Y_t,$$

where $\alpha \in \mathbb{R}^n$.

Definition 13. *The linear function $\alpha' Y_t$ is called the linear projection of X_{t+1} onto Y_t if the forecast error $(X_{t+1} - \alpha' Y_t)$ is uncorrelated with Y_t, which means that*

$$\mathbb{E}[(X_{t+1} - \alpha' Y_t) Y_t'] = 0'. \tag{3.3}$$

Theorem 4. *Let $\alpha' Y_t$ be the linear projection of X_{t+1} onto Y_t. Among all linear forecasts of X_{t+1}, the linear projection is the one with the minimal MSE.*

Proof. With similar ideas as in Theorem 3 let $g' X_t$ be an arbitrary linear forecast, whose MSE is given by

$$\begin{aligned} \mathbb{E}(X_{t+1} - g' Y_t)^2 &= \mathbb{E}(X_{t+1} - \alpha' Y_t + \alpha' Y_t - g' Y_t)^2 \\ &= \mathbb{E}(X_{t+1} - \alpha' Y_t)^2 \\ &+ 2\,\mathbb{E}\{[X_{t+1} - \alpha' Y_t][\alpha' Y_t - g' Y_t]\} \\ &+ \mathbb{E}[\alpha' Y_t - g' Y_t]^2. \end{aligned} \tag{3.4}$$

As in (3.1) the middle term of (3.4) zeros out if (3.3) holds and we get

$$\mathbb{E}([X_{t+1} - \alpha' Y_t][\alpha' Y_t - g' Y_t]) = (\mathbb{E}[X_{t+1} - \alpha' Y_t] Y_t')[\alpha - g] = 0'[\alpha - g].$$

Thus the MSE of $g' Y_t$ simplifies to

$$\mathbb{E}[X_{t+1} - g' Y_t]^2 = \mathbb{E}[X_{t+1} - \alpha' Y_t]^2 + \mathbb{E}[\alpha' Y_t - g' Y_t]^2.$$

It gets minimal if

$$g' Y_t = \alpha' Y_t$$

holds. The optimal linear forecast is therefore $\alpha' Y_t$ which fulfills condition (3.3). □

3.2 State space model of a dynamic system

There are many possibilities to define a (linear Gaussian) state space model. First, we follow [9, p. 43] and use the following form:

$$\begin{aligned} y_t &= Z_t\alpha_t + \epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0, H_t), \\ \alpha_{t+1} &= T_t\alpha_t + R_t\eta_t, \quad \eta_t \sim \mathcal{N}(0, Q_t), \qquad t = 1, ..., n, \end{aligned} \tag{3.5}$$

where y_t is a $(p \times 1)$ vector, consisting of observations called the *observation vector* and α_t is an unobservable $(m \times 1)$ vector called the *state vector*. Therefore the first equation of (3.5) is called the *observation equation* and the second is called the *state equation*. We want to describe the development of the system (3.5) depending on time t. However, because α_t is not directly observable, we need to analyse the system based on the observations y_t. We assume that the matrices Z_t, T_t, R_t, H_t and Q_t are known and that the error terms ϵ_t and η_t are serially independent and independent of each other for arbitrary time points t. The matrices Z_t and T_{t-1} can be dependent on $y_1, \ldots, y_{t-1}$. We assume that the initial state vector α_1 is $\mathcal{N}(a_1, P_1)$ distributed and independent of $\epsilon_1, \ldots, \epsilon_n$ and $\eta_1, \ldots, \eta_n$, where we assume a_1 and P_1 as known. A summary of the dimensions of (3.5) can be found in Table 3.1.

Vector		Matrix	
y_t	$(p \times 1)$	Z_t	$(p \times m)$
α_t	$(m \times 1)$	T_t	$(m \times m)$
ϵ_t	$(p \times 1)$	H_t	$(p \times p)$
η_t	$(r \times 1)$	R_t	$(m \times r)$
		Q_t	$(r \times r)$
a_1	$(m \times 1)$	P_1	$(m \times m)$

Table 3.1: Dimensions of the vectors and matrices of the state space model

3.2.1 State space model of an ARIMA process

The goal is that we put our defined models for describing our time series in state space form. It can be found in more detail in [9, pp. 53-55]. First, we want to use the ARMA(p, q) model of Definition 3, given by

$$y_t = \phi_1 y_{t-1} + \ldots + \phi_p y_{t-p} + \zeta_t + \theta_1\zeta_{t-1} + \ldots + \theta_q\zeta_{t-q}, \qquad \text{where } \zeta_t \sim \mathcal{N}(0, \sigma^2) \tag{3.6}$$

and reshape it into (3.5). We can write (3.6) as

$$y_t = \sum_{j=1}^{r} \phi_j y_{t-j} + \zeta_t + \sum_{j=1}^{r-1} \theta_j \zeta_{t-j}, \qquad t = 1, \dots, n, \tag{3.7}$$

where $r = \max\{p, q+1\}$ (some coefficients might be zero).

Remark 3. *We can also add a constant term to (3.7) to get an ARMA equation with a constant and also a constant can be added to an ARIMA equation. We will make use of this model later when creating a model in our evaluations but for reasons of simplicity, we will omit this here.*

Now we can define

$$\begin{aligned} Z_t &:= (1, 0, \dots, 0), \\ \alpha_t &:= \begin{pmatrix} y_t \\ \phi_2 y_{t-1} + \dots + \phi_r X_{t-r+2} + \theta_1 \zeta_t + \dots + \theta_{r-1} \zeta_{t-r+2} \\ \phi_3 X_{t-1} + \dots + \phi_r X_{t-r+2} + \theta_2 \zeta_t + \dots + \theta_{r-1} \zeta_{t-r+3} \\ \vdots \\ \phi_r y_{t-1} + \theta_{r-1} \zeta_t \end{pmatrix} \end{aligned} \tag{3.8}$$

and write the state equation for α_{t+1} with

$$T_t := T = \begin{bmatrix} \phi_1 & 1 & & 0 \\ \vdots & & \ddots & \\ \phi_{r-1} & 0 & & 1 \\ \phi_r & 0 & \cdots & 0 \end{bmatrix}, \qquad R_t := R = \begin{pmatrix} 1 \\ \theta_1 \\ \vdots \\ \theta_{r-1} \end{pmatrix}, \qquad \eta_t := \zeta_{t+1}. \tag{3.9}$$

This gives us together with the observation equation $y_t = Z_t \alpha_t$, the model we considered at (3.5) with $\epsilon_t = 0$, implying that $H_t = 0$.

Example 6. *Let us consider an ARMA(2,1) process. We can write it in state space form with $r = 2$ as*

$$\begin{pmatrix} y_{t+1} \\ \phi_2 y_t + \theta_1 \zeta_{t+1} \end{pmatrix} = \begin{bmatrix} \phi_1 & 1 \\ \phi_2 & 0 \end{bmatrix} \begin{pmatrix} y_t \\ \phi_2 y_{t-1} + \theta_1 \zeta_t \end{pmatrix} + \begin{pmatrix} 1 \\ \theta_1 \end{pmatrix} \zeta_{t+1}.$$

Now we want to consider an ARIMA(p, d, q) model given by $y_t^* = \nabla^d y_t$. Let us first consider the state space form of an ARIMA model with parameters $(2, 1, 1)$ which is given by

$$\begin{aligned} y_t &= (1, 1, 0)\alpha_t, \\ \alpha_{t+1} &= \begin{bmatrix} 1 & 1 & 0 \\ 0 & \phi_1 & 1 \\ 0 & \phi_2 & 0 \end{bmatrix} \alpha_t + \begin{pmatrix} 0 \\ 1 \\ \theta_1 \end{pmatrix} \zeta_{t+1} \end{aligned}$$

with the state vector

$$\alpha_t = \begin{pmatrix} y_{t-1} \\ y_t^* \\ \phi_2 y_{t-1}^* + \theta_1 \zeta_t \end{pmatrix}$$

and $y_t^* = \nabla y_t = y_t - y_{t-1}$. This example generalises to models with $d = 1$ and other values for p and q. If we choose $p = 2$, $d = 2$ and $q = 2$ the state space form is given by

$$\begin{aligned} y_t &= (1, 1, 1, 0)\alpha_t, \\ \alpha_{t+1} &= \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & \phi_1 & 1 \\ 0 & 0 & \phi_2 & 0 \end{bmatrix} \alpha_t + \begin{pmatrix} 0 \\ 0 \\ 1 \\ \theta_1 \end{pmatrix} \zeta_{t+1} \end{aligned}$$

with the state vector

$$\alpha_t = \begin{pmatrix} y_{t-1} \\ \nabla y_{t-1} \\ y_t^* \\ \phi_2 y_{t-1}^* + \theta_1 \zeta_t \end{pmatrix},$$

where $y_t^* = \nabla^2 y_t = \nabla(y_t - y_{t-1})$. The relations follow because of

$$\begin{aligned} \nabla y_t &= \nabla^2 y_t + \nabla y_{t-1}, \\ y_t &= \nabla y_t + y_{t-1} = \nabla^2 y_t + \nabla y_{t-1} + y_{t-1}. \end{aligned}$$

For a general $d > 0$, under the consideration of (c.f. [5, pp. 114-115])

$$\nabla^{d-j} y_t = y_t^* + \sum_{i=1}^{j} \nabla^{d-i} y_{t-1}, \qquad j = 1, \ldots, d,$$

we can choose the following matrices for a suitable state space model:

$$\alpha_t := \begin{pmatrix} y_{t-1} \\ \nabla y_{t-1} \\ \vdots \\ \nabla^{d-1} y_{t-1} \\ y_t^* \\ \phi_2 y_{t-1}^* + \ldots + \phi_r y_{t-r+1}^* + \theta_1 \zeta_t + \ldots + \theta_{r-1} \zeta_{t-r+2} \\ \phi_3 y_{t-1}^* + \ldots + \phi_r y_{t-r+2}^* + \theta_3 \zeta_t + \ldots + \theta_{r-1} \zeta_{t-r+3} \\ \vdots \\ \phi_r y_{t-1}^* + \theta_{r-1} \zeta_t \end{pmatrix},$$

$$Z_t := (1, 1, \ldots, 1, 0, \ldots, 0),$$

$$T_t := \begin{bmatrix} 1 & 1 & \ldots & 1 & 0 & \ldots & \ldots & 0 \\ 0 & 1 & \ldots & 1 & 0 & \ldots & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & 1 & 1 & 0 & \ldots & \ldots & 0 \\ 0 & \ldots & 0 & \phi_1 & 1 & 0 & \ldots & 0 \\ 0 & \ldots & 0 & \phi_2 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \ddots & \\ \ldots & \ldots & \ldots & \phi_{r-1} & 0 & \ldots & 0 & 1 \\ 0 & \ldots & 0 & \phi_r & 0 & \ldots & 0 & 0 \end{bmatrix} \quad \text{and}$$

$$R_t := (0, \ldots, 0, 1, \theta_1, \ldots, \theta_{r-1})'.$$

3.2.2 State space model of a VARMA process

It can be profitable, especially for vector-valued processes to write the state space model in a slightly different form, such as for example in [21, p. 613]:

$$\begin{aligned} y_t &= Z_t\alpha_t + F_t x_t + u_t, \qquad t = 1, 2, \ldots, \\ \alpha_{t+1} &= T_t\alpha_t + G_t x_t + v_t, \qquad t = 0, 1, 2, \ldots. \end{aligned} \tag{3.10}$$

Here
y_t is a $(k \times 1)$ vector, the observation vector,
α_t is an $(n \times 1)$ vector, the state vector,
x_t is a $(k \times 1)$ vector of observable inputs,
u_t is a $(k \times 1)$ vector of observation or measurement errors or noise,
v_t is an $(n \times 1)$ vector of system or transition equation errors or noise,
Z_t is a $(k \times n)$ measurement matrix,
F_t is a $(k \times m)$ input matrix of the observation equation,
T_t is an $(n \times n)$ transition matrix and
G_t is an $(n \times m)$ input matrix of the transition equation.

The matrices Z_t, F_t, T_t and G_t are assumed to be known at time t, although at least some of them will often be time-invariant. In case of a zero-mean VARMA(p, q) process (c.f. Definition 9) of the form

$$y_t = A_t y_{t-1} + \ldots + A_p y_{t-p} + \zeta_t + B_1\zeta_{t-1} + \ldots + B_q\zeta_{t-q}, \tag{3.11}$$

we can bring it in state space form of (3.10) by defining

$$\alpha_t := \underbrace{\begin{pmatrix} y_t \\ \vdots \\ y_{t-p+1} \\ \zeta_t \\ \vdots \\ \zeta_{t-q+1} \end{pmatrix}}_{[k(p+q)\times 1]}, \quad x_t := 0, \quad u_t := 0, \quad v_t := \begin{pmatrix} \zeta_{t+1} \\ 0 \\ \vdots \\ 0 \\ \zeta_{t+1} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \\ \\ \end{matrix}\right\} (kp \times 1) \\ \left.\begin{matrix} \\ \\ \\ \\ \end{matrix}\right\} (kq \times 1) \end{matrix},$$

$$Z_t := \underset{(k\times k(p+q))}{[I_k, 0_k, \ldots, 0_k]}, \quad F := 0, \quad T := \underset{[k(p+q)\times k(p+q)]}{\begin{bmatrix} A & B \\ C & D \end{bmatrix}} \quad \text{and} \quad G := 0, \quad \text{where}$$

$$A := \underset{(kp\times kp)}{\begin{bmatrix} A_1 & \ldots & A_{p-1} & A_p \\ I_k & & 0 & 0 \\ & \ddots & & \\ 0 & \ldots & I_k & 0 \end{bmatrix}}, \quad B := \underset{(kp\times kq)}{\begin{bmatrix} B_1 & \ldots & B_{q-1} & B_q \\ 0 & \ldots & 0 & 0 \\ \vdots & & \vdots & \vdots \\ 0 & \ldots & 0 & 0 \end{bmatrix}}, \quad \underset{(kq\times kp)}{C := 0} \quad \text{and}$$

$$D := \underset{(kq\times kq)}{\begin{bmatrix} 0 & \ldots & 0 & 0 \\ I_k & & 0 & 0 \\ & \ddots & & \vdots \\ 0 & \ldots & I_k & 0 \end{bmatrix}}.$$

If a constant ν is added to (3.11) we can modify the above vectors and matrices to bring it in state space form. One possibility is to set

$$x_t := 1 \quad \text{and choose} \quad G := \underset{[k(p+q)\times 1]}{\begin{pmatrix} \nu \\ 0 \\ \vdots \\ 0 \end{pmatrix}}.$$

Another way is to modify the state vector α_t as

$$\alpha_t = \underset{(k(p+q)+1\times 1)}{\begin{pmatrix} 1 \\ y_t \\ \vdots \\ y_{t-p+1} \\ \zeta_t \\ \vdots \\ \zeta_{t-q+1} \end{pmatrix}}, \quad \text{the transition vector as} \quad v_t = \underset{[k(p+q)+1\times 1]}{\begin{pmatrix} 0 \\ \zeta_{t+1} \\ 0 \\ \vdots \\ 0 \\ \zeta_{t+1} \\ 0 \\ \vdots \\ 0 \end{pmatrix}}$$

and choose the matrices A and B of the block matrix T as (T now has dimension $[k(p+q)+1 \times k(p+q)+1]$):

$$A := \underbrace{\begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ \nu & A_1 & \dots & A_{p-1} & A_p \\ 0 & I_k & & 0 & 0 \\ \vdots & \ddots & & & \\ 0 & 0 & \dots & I_k & 0 \end{bmatrix}}_{(kp+1 \times kp+1)}, \quad B := \underbrace{\begin{bmatrix} 0 & \dots & 0 & 0 \\ B_1 & \dots & B_{q-1} & B_q \\ 0 & \dots & 0 & 0 \\ \vdots & & \vdots & \vdots \\ 0 & \dots & 0 & 0 \end{bmatrix}}_{(kp+1 \times kq+1)}.$$

3.2.3 State space model of a VARMAX process

We can cast a VARMAX(p, q, b) (c.f. Definition 10) process of the form:

$$y_t = \sum_{i=1}^{p} A_i y_{t-i} + \zeta_t + \sum_{j=1}^{q} B_j \zeta_{t-j} + \sum_{j=k}^{b} M_k u_{t-k}, \quad t \in T, \tag{3.12}$$

in state space form of (3.10) by choosing the vectors and matrices as in the VARMA case and modifying the following vectors and matrices:

$$\alpha_t = \underbrace{\begin{bmatrix} y_t \\ \vdots \\ y_{t-p+1} \\ u_t \\ \vdots \\ u_{t-b+1} \\ \zeta_t \\ \vdots \\ \zeta_{t-q+1} \end{bmatrix}}_{(k(p+q+b) \times 1)}, \quad v_t = \begin{pmatrix} \zeta_{t+1} \\ 0 \\ \vdots \\ 0 \\ \zeta_{t+1} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \\ \\ \end{matrix}\right\} (k(p+r) \times 1) \\ \left.\begin{matrix} \\ \\ \\ \\ \end{matrix}\right\} (kq \times 1) \end{matrix}, \quad F = \underbrace{\begin{bmatrix} B_0 \\ 0_{k \times r} \\ \vdots \\ 0_{k \times r} \\ I_r \\ 0_{r \times r} \\ \vdots \\ 0_{r \times r} \\ 0_{k \times r} \\ \vdots \\ 0_{k \times r} \end{bmatrix}}_{(k(p+q+b) \times 1)} \quad \text{and}$$

$Z_t = [I_k, 0_{k\times k}, \ldots, 0_{k\times k}, \underbrace{0_{k\times r}, \ldots, 0_{k\times r}}_{(k\times k(p+b+q))}, 0_{k\times k}, \ldots, 0_{k\times k}]$. Finally the transition equation $T = \underbrace{\begin{bmatrix} A & M & B \\ 0_{rb\times kp} & N & 0_{rb\times kq} \\ C & 0_{kq\times rb} & D) \end{bmatrix}}_{[(k(p+q)+rb)\times k(r+b+q)]}$, where $M = \underbrace{\begin{bmatrix} M_1 & \ldots & M_{b-1} & M_b \\ 0 & \ldots & 0 & 0 \\ \vdots & & \vdots & \vdots \\ 0 & \ldots & 0 & 0, \end{bmatrix}}_{(rb\times rb)}$,

$N = \underbrace{\begin{bmatrix} 0 & \ldots & 0 & 0 \\ I_r & & 0 & 0 \\ & \ddots & & \vdots \\ 0 & \ldots & I_r & 0 \end{bmatrix}}_{(rb\times rb)}$ with the other block matrices in T as in the VARMA case.

To add a constant ν to the VARMAX model (3.12) we can use the tools we discussed in the VARMA state space model.

3.2.4 State space model of regression models

Considering the regression model of Definition 12 of the form

$$y_t = X_t\beta + \xi, \quad t = 1, \ldots, n,$$

where ξ follows an ARMA model (2.1). We can easily bring it in state space form of (3.5) by choosing (c.f. [9, pp. 60-61]) α_t as in (3.8) and let

$$\alpha_t^* := \begin{pmatrix} \beta \\ \alpha_t \end{pmatrix}, \quad T^* := \begin{bmatrix} I_k & 0 \\ 0 & T \end{bmatrix}, \quad R^* := \begin{bmatrix} 0 \\ R \end{bmatrix} \quad Z_t^* := (X_t, 1, 0, \ldots, 0)',$$

where T and R are defined as in (3.9). Then the model

$$\begin{aligned} y_t &= Z_t^*\alpha_t, \\ \alpha_{t+1}^* &= T^*\alpha_t^* + R^*\eta_t \end{aligned}$$

is in state space form.

3.3 The Kalman filter

The Kalman filter can be used for smoothing, filtering and forecasting a time series. Once we put our model in state space form we can use the recursive procedure for computing the optimal estimator of the state vector at time t. Optimal in the sense of the MSE, i.e. the conditional expectation of α_t based on a set of past observations Y_t (c.f. Theorem 3) available at time t. We follow [9, pp. 82-85] and consider the linear state space model of (3.5). However, all of the following techniques also exist for the state space model of (3.10) and therefore they can also be applied to our vector-valued forecasting methods (c.f. [21, pp. 626-631]). Our goal is to build up the distributions of α_t and y_t recursively with the initial state α_1 which is $\mathcal{N}(a_1, P_1)$ distributed and a_1 and P_1 are known. We indicate Y_{t-1} as the set of past observations of $y_1, \dots y_{t-1}$, $t = 2, 3, \dots$ while Y_0 indicates that there is no prior observation before $t = 1$. We need to keep in mind that
$p(y_t|\alpha_1, \dots, \alpha_t, Y_t) = p(y_t|\alpha_t)$ and $p(\alpha_t|\alpha_1, \dots, \alpha_t, Y_t) = p(\alpha_{t+1}|\alpha_t)$. Let

$$\begin{aligned} a_{t|t} &:= \mathbb{E}(\alpha_t|Y_t), \\ a_{t+1} &:= \mathbb{E}(\alpha_{t+1}|Y_t), \\ P_{t|t} &:= \mathrm{Var}(\alpha_t|Y_t) \quad \text{and} \\ P_{t+1} &:= \mathrm{Var}(\alpha_{t+1}|Y_t). \end{aligned}$$

We call $a_{t|t}$ as the filtered estimator of the state α_t and a_{t+1} as the 1-step ahead predictor of α_{t+1} with associated variances $P_{t|t}$ and P_{t+1}. Since all distributions are normally distributed, it follows that conditional distributions of subsets of variables given other subsets of variables are also normally distributed. Therefore the distributions α_t given Y_t and α_{t+1} given Y_t are given by

$$\begin{aligned} p(\alpha_t|Y_t) &= \mathcal{N}(a_{t|t}, P_{t|t}) \quad \text{and} \\ p(\alpha_{t+1}|Y_t) &= \mathcal{N}(a_{t+1}, P_{t+1}). \end{aligned}$$

We start with our calculations inductively starting with $p(\alpha_t|Y_{t-1}) = \mathcal{N}(a_t, P_t)$. We want to calculate $a_{t|t}$, a_{t+1}, P_{t+1} from a_t and P_t recursively for $t = 1, \dots, n$. Let

$$v_t = y_t - \mathbb{E}(y_t|Y_{t-1}) = y_t - \mathbb{E}(Z_t\alpha_t + \epsilon_t|Y_{t-1}) = y_t - Z_t a_t.$$

Therefore v_t is the 1-step ahead forecast error of y_t given Y_{t-1} and $\mathbb{E}(\alpha_t|Y_t) = \mathbb{E}(\alpha_t|Y_{t-1}, v_t)$ because when Y_{t-1} and v_t are fixed then Y_t is fixed and vice versa.
It holds that $\mathbb{E}(v_t|Y_{t-1}) = \mathbb{E}(y_t - Z_t a_t|Y_{t-1}) = \mathbb{E}(Z_t\alpha_t + \epsilon_t - Z_t a_t|Y_{t-1}) = 0$. Thus $\mathbb{E}(v_t) = 0$ and $\mathrm{Cov}(y_j, v_t) = \mathbb{E}[y_j\,\mathbb{E}(v_t|Y_{t-1})'] = 0$ for $j = 1, \ldots, t-1$. Now we need the following Lemma (c.f. [9, pp. 77-78]).

Lemma 1. *Suppose that x, y are jointly normally distributed random vectors with*

$$\mathbb{E}\begin{pmatrix} x \\ y \end{pmatrix} := \begin{pmatrix} \mu_x \\ \mu_y \end{pmatrix}, \quad \mathrm{Var}\begin{pmatrix} x \\ y \end{pmatrix} := \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{xy} & \Sigma_{yy} \end{bmatrix}.$$

Then the conditional distribution of x given y is normal with mean vector

$$\mathbb{E}(x|y) = \mu_x + \Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y), \tag{3.13}$$

and variance matrix

$$\mathrm{Var}(x|y) = \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{xy}'. \tag{3.14}$$

Proof. Let $z = x - \Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y)$. The transformation from (x, y) to (y, z) is linear and (x, y) is normally distributed, the joint distribution of y and z is also normally distributed. It follows that

$$\begin{aligned} \mathbb{E}(z) &= \mu_x, \\ \mathrm{Var}(z) &= \mathbb{E}[(z - \mu_x)(z - \mu_x)'] \\ &= \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{xy}', \\ \mathrm{Cov}(y, z) &= \mathbb{E}[y(z - \mu_x)'] \\ &= \mathbb{E}[y(x - \mu_x)' - y(y - \mu_y)'\Sigma_{yy}^{-1}\Sigma_{xy}'] \\ &= 0. \end{aligned} \tag{3.15}$$

We conclude that z is distributed independently of y because they are normal and uncorrelated. Also, the distribution of z does not depend on y, its conditional distribution given y is the same as its unconditional distribution. That means it is normal with mean vector μ_x and variance matrix (3.15) which is the same as (3.14). Because $z = x - \Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y)$, it follows that the conditional distribution of x given y is normal with mean vector (3.13) and variance matrix (3.14). □

Applying Lemma 1 to the conditional joint distribution of α_t and v_t given Y_{t-1}, taking x and y in Lemma 1 as α_t and v_t here, it follows

$$a_{t|t} = \mathbb{E}(\alpha_t|Y_{t-1}) + \mathrm{Cov}(\alpha_t, v_t)[\mathrm{Var}(v_t)]^{-1}v_t,$$

where Cov and Var refer to covariance and variance in the conditional joint distributions of α_t and v_t given Y_{t-1}. It holds by definition of P_t that

$$\begin{aligned}\mathrm{Cov}(\alpha_t, v_t) &= \mathbb{E}[\alpha_t(Z_t\alpha_t + \epsilon_t - Z_t a_t)'|Y_{t-1}] \\ &= \mathbb{E}[\alpha_t(\alpha_t - a_t)'Z_t|Y_{t-1}] \\ &= P_t Z_t'.\end{aligned}$$

Let

$$F_t = \mathrm{Var}(v_t|Y_{t-1}) = \mathrm{Var}(Z_t\alpha_t + \epsilon_t - Z_t a_t|Y_{t-1}) = Z_t P_t Z_t' + H_t, \tag{3.16}$$

then it holds that

$$a_{t|t} = a_t + P_t Z_t' F_t^{-1} v_t. \tag{3.17}$$

Applying (3.14) of Lemma 1 we have, assuming that F_t is non-singular (this assumption is normally valid in well-formulated models):

$$\begin{aligned}P_{t|t} = \mathrm{Var}(\alpha_t|Y_t) &= \mathrm{Var}(\alpha_t|Y_{t-1}, v_t) \\ &= \mathrm{Var}(\alpha_t|Y_{t-1}) - \mathrm{Cov}(\alpha_t, v_t)[\mathrm{Var}(v_t)]^{-1}\mathrm{Cov}(\alpha_t, v_t)' \\ &= P_t - P_t Z_t' F_t^{-1} Z_t P_t. \end{aligned} \tag{3.18}$$

The relations (3.17) and (3.18) are sometimes called the *updating step* of the Kalman filter. Now we want to develop recursions for a_{t+1} and P_{t+1}.
Because $\alpha_{t+1} = T_t\alpha_t + R_t + \eta_t$, we have

$$\begin{aligned}\alpha_{t+1} &= \mathbb{E}(T_t\alpha_t + R_t\eta_t|Y_t) \\ &= T_t\,\mathbb{E}(\alpha_t|Y_t), \end{aligned} \tag{3.19}$$

$$\begin{aligned}P_{t+1} &= \mathrm{Var}(T_t\alpha_t + R_t\eta_t|Y_t) \\ &= T_t\mathrm{Var}(\alpha_t|Y_t)T_t' + R_t Q_t R_t' \quad \text{for } t = 1, \ldots, n. \end{aligned} \tag{3.20}$$

It holds, substituting (3.17) into (3.19)

$$a_{t+1} = T_t a_{t|t} = T_t a_t + K_t v_t, \quad t = 1, \dots n, \tag{3.21}$$

where

$$K_t = T_t P_t Z_t' F_t^{-1}. \tag{3.22}$$

The matrix K_t is often called *Kalman gain*. With this approach, a_{t+1} has been obtained as a linear function of the previous value a_t and the forecast error v_t of y_t given Y_{t-1}. If we substitute from (3.18) and (3.22) in (3.20) we get

$$P_{t+1} = T_t P_t (T_t - K_t Z_t)' + R_t Q_t R_t', \quad t = 1, \dots, n. \tag{3.23}$$

The relations (3.21) and (3.23) are often called the *prediction step* of the Kalman filter. With the recursions (3.17), (3.21), (3.18) and (3.23) we now constituted the celebrated Kalman filter for the state space model of (3.5). An advantage is that they enable us to update our knowledge of the system each time a new observation comes in and we do not have to invert a $(pt \times pt)$ matrix to fit the model each time the t-th observation comes in for $t = 1, \dots, n$. We only have to invert the $(p \times p)$ matrix F_t and p is generally much smaller than n. We now collect together the filter equations:

$$\begin{aligned} v_t &= y_t - Z_t a_t, & F_t &= Z_t P_t Z_t' + H_t, \\ a_{t|t} &= a_t + P_t Z_t' F_t^{-1} v_t, & P_{t|t} &= P_t - P_t Z_t' F_t^{-1} Z_t P_t, \\ a_{t+1} &= T_t a_t + K_t v_t, & P_{t+1} &= T_t P_t (T_t - K_t Z_t)' + R_t Q_t R_t' \end{aligned} \tag{3.24}$$

for $t = 1, \dots, n$, where $K_t = T_t P_t Z_t' F_t^{-1}$ with a_1 and P_1 which are known as well as the mean vector and variance matrix of the initial state vector α_1. The recursion of (3.24) is called the *Kalman filter*. Once we computed $a_{t|t}$ and $P_{t|t}$, it is sufficient to adopt the relations

$$a_{t+1} = T_t a_{t|t}, \quad P_{t+1} = T_t P_{t|t} T_t' + R_t Q_t R_t',$$

for predicting the state vector α_{t+1} and its variance matrix at time t. The dimensions of the used vectors and matrices can be seen in Table 3.2.

Vector		Matrix	
v_t	$(p \times 1)$	F_t	$(p \times p)$
		K_t	$(m \times p)$
a_t	$(m \times 1)$	P_t	$(m \times m)$
$a_{t\|t}$	$(m \times 1)$	$P_{t\|t}$	$(m \times m)$

Table 3.2: Dimensions of the vectors and matrices of the Kalman filter

3.3.1 Maximum likelihood estimation of parameters

So far we assumed that the parameters of the forecast models are known. We want to give an overview of how we can calculate the likelihood to estimate the parameters of a forecast model that we put in a state space model (3.5) based on [9, chapter 7]. We shall show that the likelihood can be calculated by a routine application of the Kalman filter. The likelihood of the vector consisting of n observations $Y_n = (y_1, \dots, y_n)'$ is

$$L(Y_n) = p(y_1, \dots y_n) = p(y_1) \prod_{t=2}^{n} p(y_t|Y_{t-1}),$$

where $Y_t = (y_1, \dots y_t)'$. We generally work with the log-likelihood

$$\log L(Y_n) = \sum_{t=1}^{n} = \log p(y_t|Y_{t-1}), \tag{3.25}$$

where $p(y_1|Y_0) = p(y_1)$. In the case of the state space model (3.5) it holds that $\mathbb{E}(y_t|Y_{t-1}) = Z_t a_t$. Now we set $v_t = y_t - Z_t a_t,\ F_t = \mathrm{Var}(y_t|Y_{t-1})$ and substitute $\mathcal{N}(Z_t a_t, F_t)$ for $p(y_t|Y_{t-1})$ in (3.25). We obtain

$$\log L(Y_n) = -\frac{np}{2} \log 2\pi - \frac{1}{2}(\log |F_t| + v_t' F_t^{-1} v_t).$$

We can easily compute the quantities v_t and F_t by the Kalman filter. In a usual situation at least some elements of the system matrices Z_t, H_t, T_t, R_t depend on a vector ψ of unknown parameters that we want to estimate. In practice, the respective forecast model is first formulated in a state space model. Then we can calculate the Log-likelihood $\log L(Y_n|\psi)$ of the model which depends on the unknown parameters ψ. Numerical maximisation algorithms are used to calculate them.

3.3.2 Information criteria

Information criteria are referenced as a measure to rate a statistical model. We want to formulate information criteria based on the calculated log-likelihood value. The goal is to measure the fit of the model given the estimated parameter vector $\hat{\psi}$. When dealing with competing models we want to compare the log-likelihood value of a particular fitted model, denoted by $\log L(Y_n|\hat{\psi})$ with the corresponding log-likelihood values of competing models (c.f. [9, pp. 187-188]). The larger the number of parameters that a model contains the larger its log-likelihood. The information criteria typically take the general form

$$\nabla_{n,k} = -2 \log L(Y_n|\hat{\psi}) + C_{n,k}, \tag{3.26}$$

where n is the sample size and k is the dimension of $\hat{\psi}$. The value of $C_{n,k}$ represents the penalty for adding parameters to the model. In our future analysis, we will use the Akaike Information Criterion (AIC) to define the parameters for our forecasting model. It is given by choosing $C_{n,k} := 2k$ in (3.26). In general, a model with a smaller AIC value is preferred.

3.3.3 Forecasts based on a state space model

In this section, we will show how to compute minimum squared error forecasts obtained by treating future values of y_t as missing observations (c.f. [9, chapter 4]). We suppose that we have given a vector of observations $y_1, \ldots, y_n$ which follow the state space model (3.5) and we wish to forecast y_{n+j} for $j = 1, \ldots, J$. For this purpose we choose the estimate $\overline{y}_{n+j}$ of y_{n+j} which has a minimum mean squared error matrix given $Y_n = (y_1, \ldots, y_n)$, that means, $\overline{F}_{n+j} = \mathbb{E}[(\overline{y}_{n+j} - y_{n+j})'Y_n]$ is a minimum in the matrix sense for all estimates of y_{n+j}. We discussed in Section 3.1.1 that the mean squared error forecast of y_{n+j} given Y_n is the conditional mean $\overline{y}_{n+j} = \mathbb{E}(Y_{n+j}|Y_n)$. For a 1-step forecast we have $y_{n+1} = Z_{n+1}\alpha_{n+1} + \epsilon_{n+1}$, so

$$\begin{aligned} \overline{y}_{n+j} &= Z_{n+1}\, \mathbb{E}(\alpha_{n+1}|Y_n) \\ &= Z_{n+1}\alpha_{n+1}, \end{aligned}$$

where a_{n+1} is the estimate (3.21) produced by the Kalman filter. Also the conditional

mean squared error matrix

$$\begin{aligned}\overline{F}_{n+1} = \overline{F}_{n+j} &= \mathbb{E}[(\overline{y}_{n+j} - y_{n+j})'Y_n] \\ &= Z_{n+1} + P_{n+1}Z'_{n+1} + H_{n+1},\end{aligned}$$

is produced by the Kalman filter relation (3.16). Now let $\overline{a}_{n+j} = \mathbb{E}(\alpha_{n+j}|Y_n)$ and $\overline{P}_{n+j} = \mathbb{E}[(\overline{a}_{n+j} - \alpha_{n+j})'Y_n]$. Because of the observation equation it holds that $y_{n+j} = Z_{n+j}\alpha_{n+j} + \epsilon_{n+j}$ and it follows that

$$\begin{aligned}\overline{y}_{n+j} &= Z_{n+j}\,\mathbb{E}(\alpha_{n+j}|Y_n) \\ &= Z_{n+j}\overline{a}_{n+j},\end{aligned}$$

with conditional mean squared error matrix

$$\begin{aligned}\overline{F}_{n+j} &= \mathbb{E}\{[Z_{n+j}(\overline{a}_{n+j} - \alpha_{n+j}) - \epsilon_{n+j}][Z_{n+j}(\overline{a}_{n+j} - \alpha_{n+j}) - \epsilon_{n+j}]'|Y_n\} \\ &= Z_{n+j}\overline{P}_{n+j}Z'_{n+j} + H_{n+j}.\end{aligned}$$

We have $\alpha_{n+j+1} = T_{n+j}\alpha_{n+j} + R_{n+j}\eta_{n+j}$ because of the state equation,

$$\begin{aligned}\overline{a}_{n+j+1} &= T_{n+j}\,\mathbb{E}(\alpha_{n+j}|Y_n) \\ &= T_{n+j}\overline{a}_{n+j},\end{aligned}$$

for $j = 1, \ldots, J-1$ with $\overline{a}_{n+1} = a_{n+1}$ and

$$\begin{aligned}\overline{P}_{n+j+1} &= \mathbb{E}[(\overline{a}_{n+j+1} - \alpha_{n+j+1})(\overline{a}_{n+j+1} - \alpha_{n+j+1})'|Y_n] \\ &= T_{n+j}\,\mathbb{E}[(\overline{a}_{n+j} - \alpha_{n+j})(\overline{a}_{n+j+1} - \alpha_{n+j+1})'|Y_n]T'_{n+j} \\ &+ R_{n+j}\,\mathbb{E}[\eta_n + \eta'_{n+j}]R'_{n+j} \\ &= T_{n+j}\overline{P}_{n+j}T'_{n+j} + R_{n+j}Q_{n+j}R'_{n+j} \quad \text{for } j = 1, \ldots, J-1.\end{aligned}$$

The recursions for $\overline{a}_{n+j}$ and $\overline{P}_{n+j}$ are the same as the recursions for a_{n+j} and P_{n+j} of the Kalman filter (3.24) provided if we take $Z_{n+j} = 0$ for $j = 1, \ldots, J-1$.

4 Evaluations

The data of the three switches we want to analyse is from switch engines installed at operational switches in the Netherlands, recorded with the Dutch construction and rail company Strukton Rails' POSS measurement system. Strukton Rail monitors over 2000 switches using the POSS system and approximately 80% of the monitored switches in the Netherlands have similar engines (NSE type engine) to the ones we are analysing [10]. The system measures the engine current (proportional to the engine power consumption) during the switch blades movement [16, p. 2]. During each attempt of an electrical motor to move the switch blades from one position to the other, the electrical current is measured and captured with a bipolar AC/DC current clamp to digitise the measurement with 50Hz by the POSS system in amperes [10]. The current measurements of the first two switches we want to examine (Switch 1 and Switch 2) have been recorded over several years as well as other relevant data such as the actual temperature measured outside of a relay house nearby where the data got sent to when a switch movement has taken place. In the chapter of Switch 1, we want to explain the discussed forecasting methods in terms of the given data basis of the railway switch and see how the switch behaves before a failure and how maintenance affects the switch, whereas in the analysis of the data of Switch 2 we want to compare the different forecasting methods against each other and see how we can best model the energy consumption of the railway switch. The data of Switch 3 contains measurements only over a few months but also contains more relevant data such as the temperature directly below the point machine or the relative humidity inside the point machine. We want to examine which variables of the data of Switch 3 have the most impact on the energy consumption and how we might improve the forecasting models. The target variable we want to forecast is always the (discrete) total power consumed by the electrical engine of the switch to move the switch blades from one position to the other. When choosing a model for a time series it is common practice to split the

data into a training set and a test set. The training set is used to create an appropriate model for the considered time series and for estimating any parameters of a forecasting method. The test set is used to evaluate the forecast. As an appropriate measure for the forecasts, we will use the sample mean squared error (rounded to one decimal place)

$$\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x_i})^2,$$

where $x_1, \ldots, x_n$ are the true values in the test set, $\hat{x_1}, \ldots, \hat{x_n}$ are the respective predicted values and n is the number of forecasting steps. To test the model we will perform 1-step, 10-step and 50-step forecasts. Box and Jenkins recommend at least 50 but preferably 100 or more observations to create a suitable model [3, p. 15]. According to Hyndman and Athanasopoulos the size of the training set is typically about 80% of the total sample but strongly dependent on how long the sample is and how far ahead we want to forecast [19, p. 77]. Also, other characteristics such as what questions we want to answer and how the data is structured are important. In the following evaluations, we will always use (unless otherwise stated) 100 observations before a test set as the corresponding training set. We have obtained similar results with forecasts based on models created using the smallest AIC with training sets consisting of the last 50, 150 and 200 observations before the respective test set (c.f. Appendix A).
The forecasting methods we have defined in Chapter 2 are based on whole numbers and do not take the different time intervals between two switch movements into account. We are only considering one switch movement after the other without considering the time index of the respective training and test set when creating a model or forecasting. We will discuss some problems that arise therefrom later on. However, for visualisation purposes, the indexes will be kept and forecasts will be indexed with the same index as the respective test set. The forecasts and tests were made using the Python Packages Statsmodels [29], scikit-learn [27] and SciPy [30].

4.1 Switch 1

The data of the recorded switch movements of Switch 1 consists of 10 variables (MeasurementID, MeasurementDirection, MeasurementTemperature, Measure-

mentDateTime, TotalPower, NumberOfEngines, Engine, DataStart, DataEnd, Data) and 21585 observations, raised from 01-01-2012 to 07-02-2017. The variables of interest are "MeasurementDirection" which is either 0 or 1, depending on which direction the engine turns the switch blades, "MeasurementTemperature" which is the temperature measured at a relay house of the switch nearby, "MeasurementDateTime" which is the time when the switch movement has taken place, "Data" which consists of the measurements of the electrical current of the engine measured with 50Hz in amperes during a switch movement and the target variable "TotalPower" (Figure 4.1) which models the total power consumption of a switch movement. In

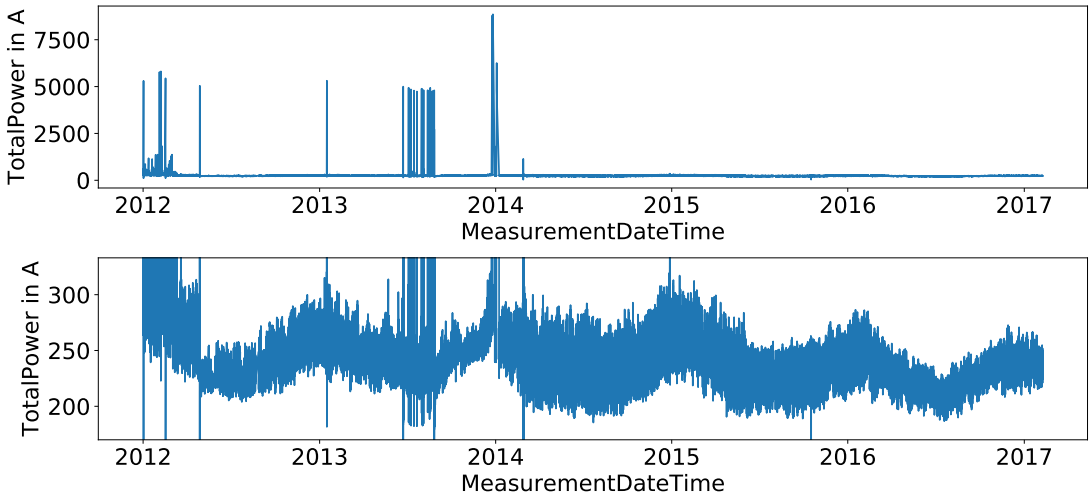

Figure 4.1: The time series TotalPower over time (top) and in enlarged view (bottom)

our future analysis we will always keep in mind that the target variable of the time series TotalPower depends on the measurement direction in which the engine turns the switch blades (Figure 4.2). This is why we might split up the original time series in two or use vector-valued models such as VARMA. In our analysis, we will divide the time series of TotalPower into 5 smaller parts. We have 13 failures, 27 repairs and 45 maintenance listings of the switch, given by the exact times of the reported failures and repairs as well as the days the maintenance has taken place, illustrated in Figure 4.3. We need enough data points between those listings for choosing the sections for the training sets with which we create a suitable model. Therefore we choose parts of the time series, where no failures, reported maintenance or repairs occurred. Also, we are not interested in failures that are certainly not predictable (e.g. the cause of an error on 23-06-2013 was a foreign object which was removed

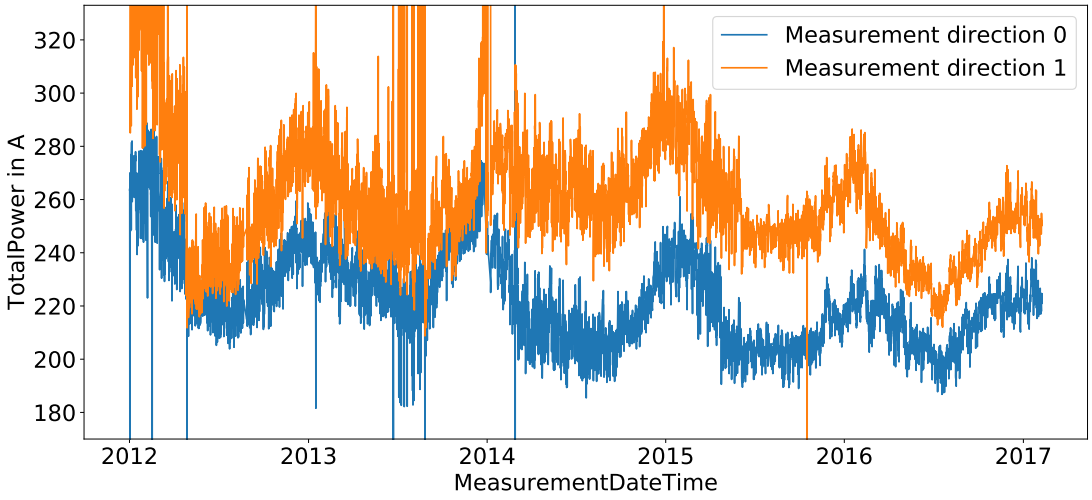

Figure 4.2: TotalPower over time, split in two directions

subsequently). We call the sets we want to analyse "data sets". The respective training and test sets lie in those sets, if not expressly mentioned the last 50 data points of a data set form the individual test set (our goal is to do a maximum of 50-step forecasts) and the last 100 data points of a data set before the test set define the individual training set. A summary of the data sets can be seen in Table 4.1 (we name them from 1 to 5 as well as their training and test sets, sorted by date). They are colourised in Figure 4.3. We want to analyse two data sets before a failure

Data set	1	2	3	4	5
Time frame	from 28-02-2013 08:16:36 to 07-04-2013 00:40:53	from 28-11-2013 14:21:57 to 26-12-2013 06:23:02	from 09-07-2014 05:51:59 to 08-08-2014 23:40:28	from 29-10-2014 10:31:16 to 28-11-2014 05:36:59	from 13-05-2015 00:30:51 to 07-06-2015 07:51:59
Number of data points	472	389	327	338	257

Table 4.1: The data sets we want to analyse of the first switch

occurred (data set 1 and 2). Therefore the last data point of the data sets 1 and 2 is the last data point before the respective failure happened. The various data sets contain all values before the last time point back to the closest event before (maintenance, repair, or failure). We only need a maximum of 150 data points to train

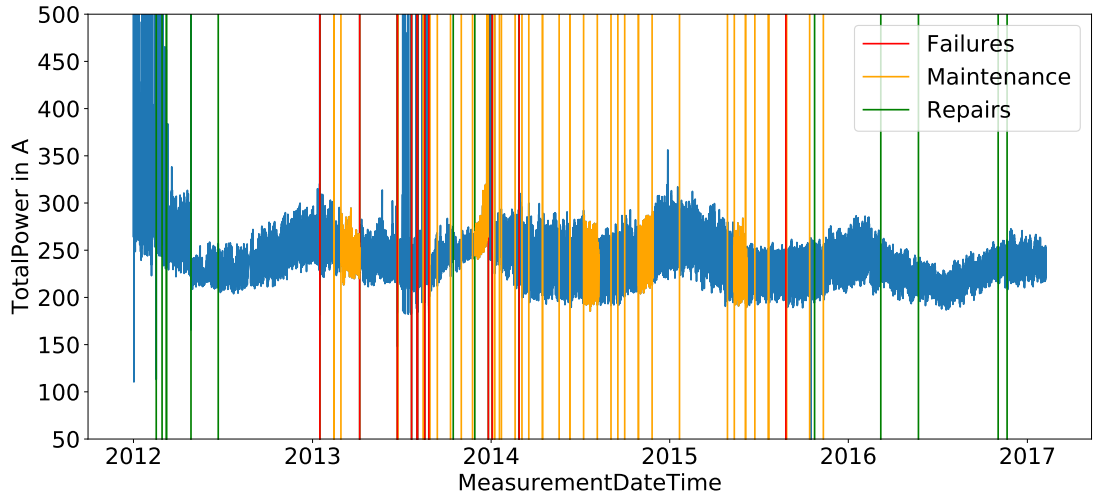

Figure 4.3: Failures, the start of the reported maintenance days, reported repairs (some lines are overlapping) and the sections we want to analyse, colourised in orange

and predict the model but with more data points not close to a time point where a failure repair or maintenance occurred we still have the possibility to shift the training and test horizon a little if the switch was already not functioning as it should. For example, it could be the case that the switch is not behaving well 150 data points before a failure and we use those biased values to create a model. That would be not expedient. The goal is to create a model that describes the total power of a functioning switch. Based on this model we want to forecast the true values of the test set. We want to investigate if the total power consumption of the switch before the failure occurred was not behaving as it should or if the failure happened without any presage. If we find a suitable model whose forecasts do not behave like the test set we can conclude that failure might happen soon and maintenance should be done.

For the other three data sets, we set the last data point 25 data points after the day maintenance has taken place. It follows that the test set starts 25 data points before the day of the maintenance and ends 25 data points after maintenance has taken place. We choose the beginning of the data sets as we did before. The goal is to see if the created models are capable of detecting maintenance of the switch that changed how the total power behaves. The maintenance that took place in the 4th data set was only an inspection unlike the ones of the 3rd and 5th data set. In this

case, the 50-step prediction of the 4th test set should be way better compared to the other ones if the model can describe the time series of TotalPower well. The first test set consists of 472 data points. As already mentioned we use the last 50 data points before the failure was reported as the test set and the training set is consisting of 100 data points before the test set (c.f. Figure 4.4).

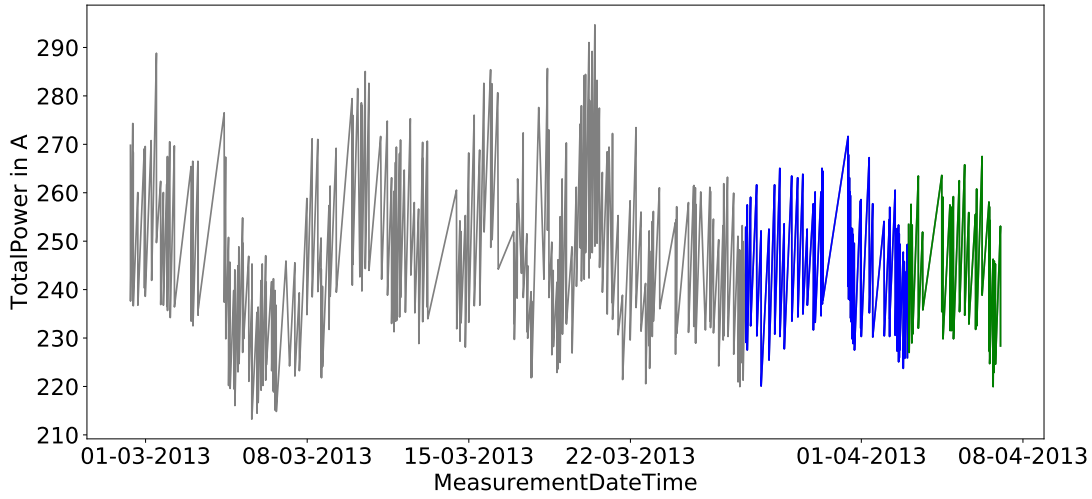

Figure 4.4: Training set (blue), test set (green) and the other data points which we do not consider (grey) of the first data set

4.1.1 ARIMA and VARMA models

If we look at the sample ACF and PACF of the training set of the first data set (Figure 4.5), we cannot be sure that we can model it as an AR(p) or MA(q) process as in our simulated case in Figure 2.1, although we can see the alternating dependence of TotalPower on the measurement direction in the sample ACF. With the information that TotalPower is dependent on the measurement direction, it would be reasonable to choose $d = 1$ for the ARIMA(p, d, q) model. Nevertheless, we will choose d using the ADF test and also based on the best AIC (those approaches should also result in choosing d as 1 but other values of d may be advantageous). First, we want to test whether the training set is stationary according to the augmented Dickey-Fuller test to choose the parameter d. We want to reject the null hypothesis to a level of $\alpha = 0.05$. As a result of the augmented Dickey-Fuller unit root test[1] we get the

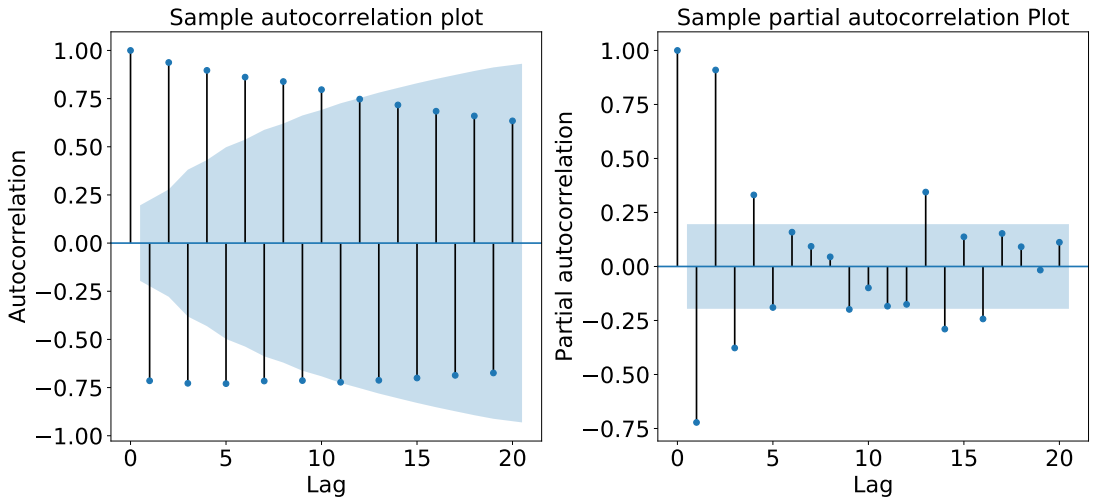

Figure 4.5: Sample ACF and PACF of the training set of the first data set

following test statistic and p-value for the first training set:

```
ADF = -1.8007798116064873
p-value = 0.380074607575997.
```

As a result, we cannot reject the null hypothesis and we cannot conclude that the time series is stationary. However, if we difference the training series, we can eliminate the alternating periodicity of the switch movement. As a result, we can test the differenced training set again and get the following test result:

```
ADF = -8.118885817485182
p-value = 1.1705923420122609e-12.
```

Therefore we can reject the null hypothesis that there is a unit root and conclude that the differenced training set is stationary. As a consequence, we choose $d = 1$. Then we use the AIC value and perform a grid search. We calculate the AIC for all possible parameters from 0 to 4 for p, d and q and save the ranked parameters and the corresponding AIC. Afterwards, we look at the sorted list of the AICs and their corresponding parameters. We now search in the list of the AICs for the computed tuple with the smallest AIC with the same value for d as we calculated. The following output shows the 3 best-calculated AIC values and their corresponding parameters:

[1] Using tsa.stattools.adfuller of Statsmodels

```
3 smallest AICs:
AIC              Parameter
560.242680       (2, 1, 2)
560.273911       (1, 1, 2)
560.613276       (4, 1, 1).
```

We see that the parameter d in the list of the smallest AICs coincides with the method of how we used to calculate d. If this is not the case in further evaluations we will also perform a forecast based on the model created with the parameters of the best AIC and add it to the evaluations. Now we fit the model with the ARIMA parameters (2,1,2). As a result, we end up with the values seen in Table 4.2.

	coef	**std err**	**z**	**P$>$ |z|**	**[0.025**	**0.975]**
const	-0.0471	0.154	-0.305	0.761	-0.350	0.255
ar.L1.D.TotalPower	-0.5001	0.270	-1.851	0.067	-1.029	0.029
ar.L2.D.TotalPower	0.4964	0.269	1.847	0.068	-0.030	1.023
ma.L1.D.TotalPower	-0.1259	0.256	-0.492	0.624	-0.627	0.375
ma.L2.D.TotalPower	-0.4756	0.113	-4.220	0.000	-0.697	-0.255

Table 4.2: Part of the summary of the fitted ARIMA(2,1,1) model of the first training set using Statsmodels

We can see the estimated constant term and the AR and MA coefficients of the differenced time series ∇X_t of the fitted ARIMA(2,1,2) process in the first column. The estimated ARIMA(2,1,2) process has the following form:

$$\nabla X_t = -0.0471 - 0.5001\nabla X_{t-1} + 0.4964\nabla X_{t-2} + Z_t - 0.1259Z_{t-1} - 0.4756Z_{t-2}.$$

The standard errors of the predicted values can be found in the second column whereas the column "z" represents the standardised coefficient (coeff/stderr). The column "P$>$ |z|" represents the p-value of the respective coefficient under the null hypothesis H_0 : coeff $= 0$. The last two columns of Table 4.2 represent the confidence interval of the estimated coefficient. We can now forecast the time series, illustrated in Figure 4.6 and calculate the mean squared errors of the forecasts:

```
Mean squared error of the 1-step forecast:  37.3
Mean squared error of the 10-step forecast:  66.4
Mean squared error of the 50-step forecast:  90.2.
```

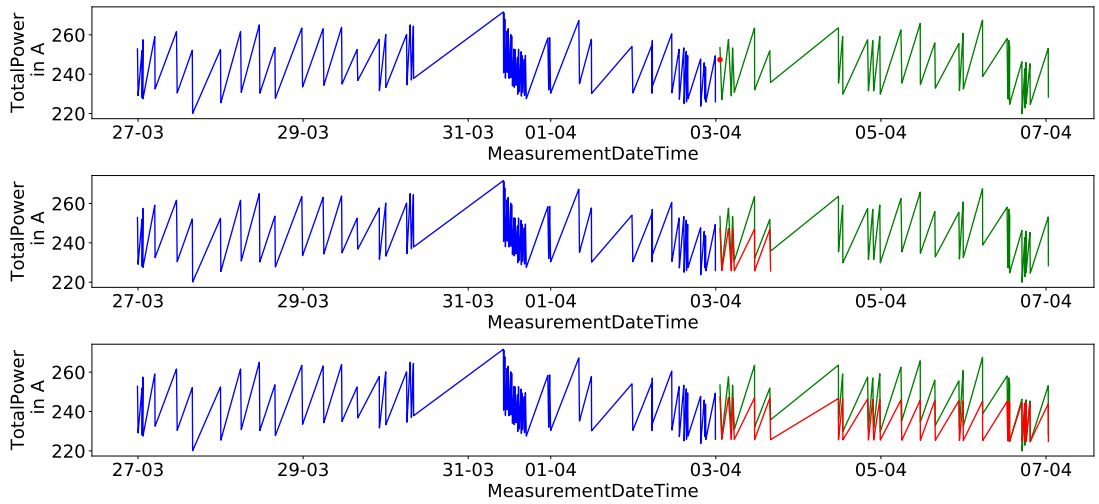

Figure 4.6: Training set (blue), test set (green) and the 1-step, 10-step and 50-step forecasts (red) of the first test set

The residuals $\epsilon_t = x_t - \hat{x}_t$ should be white noise if the model fits the data. After a look at the 50-step forecast in Figure 4.6 and at the histogram in Figure 4.7 we see that the forecast is clearly underestimating the true values of the test set.

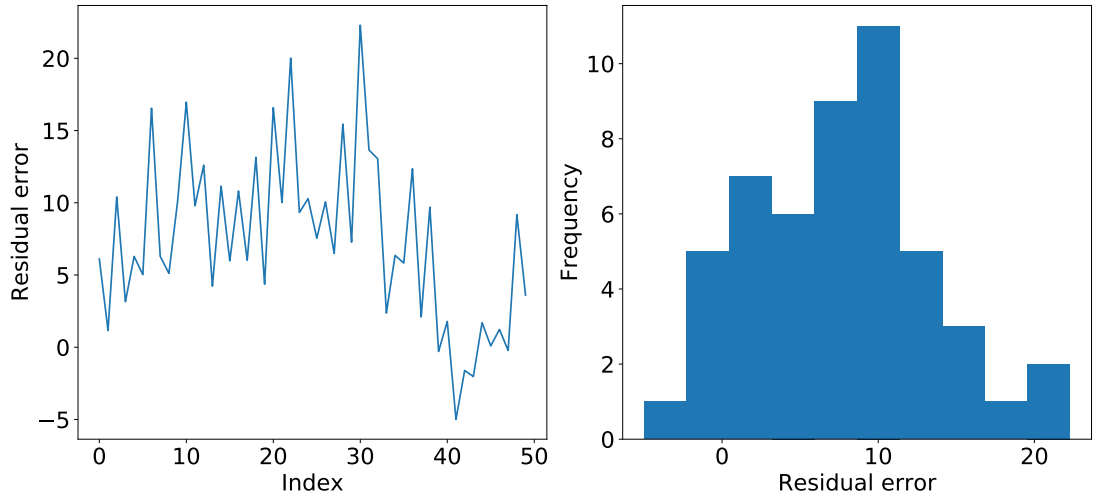

Figure 4.7: A plot (left) and histogram (right) of the residuals

We will follow the same procedure for the next data sets. For the second data set we had to difference the training set 2 times to reject the null hypothesis of the

augmented Dickey-Fuller test. However, in the sorted list, the model parameters of the best AIC are (4,1,0) and the model with the smallest AIC where $d = 2$ is (3,2,0) in eighth place. We calculate forecasts with both models and look at their MSEs((3,2,0)/(4,1,0)):

```
Mean squared error of the 1-step forecast:  1.0/18.1
Mean squared error of the 10-step forecast:  236.6/91.2
Mean squared error of the 50-step forecast:  2862896.2/2907829.
```

With both parameters, we get a big MSE for the 50-step forecast. The error is big because we have 2 outliers in the test set (Figure 4.8). Also the alternating behaviour of the measurement direction changes.

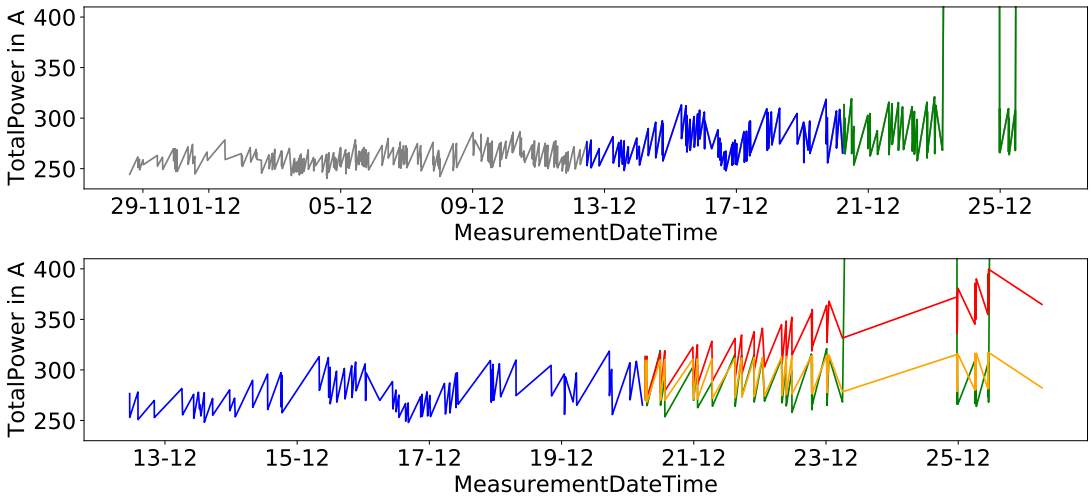

Figure 4.8: Second data set. Training set (blue) and test set (green). At the top the data we do not consider (grey). The ARIMA(3,2,1) forecast (red) and the ARIMA(4,1,0) forecast (orange) at the bottom

We conclude that the failure has happened before it has been reported. We see that those forecasts are unusable. Therefore we will not calculate any more forecasts of this test set based on this training set. Instead, we shift the training and test set 80 data points to the left. This gives us the advantage that the outliers are not in the shifted test set and the shifted training set ends on the 12-12-2013 where TotalPower starts to rise and fall (Figure 4.9). We want to answer the question if this happens due to abnormal behavior or it is still a normal action. The MSEs of

the estimated ARIMA(3,0,4) forecast can be seen below and the 50-step forecast is illustrated at the bottom of Figure 4.9:

```
Mean squared error of the 1-step forecast:  41.2
Mean squared error of the 10-step forecast:  154.5
Mean squared error of the 50-step forecast:  278.8.
```

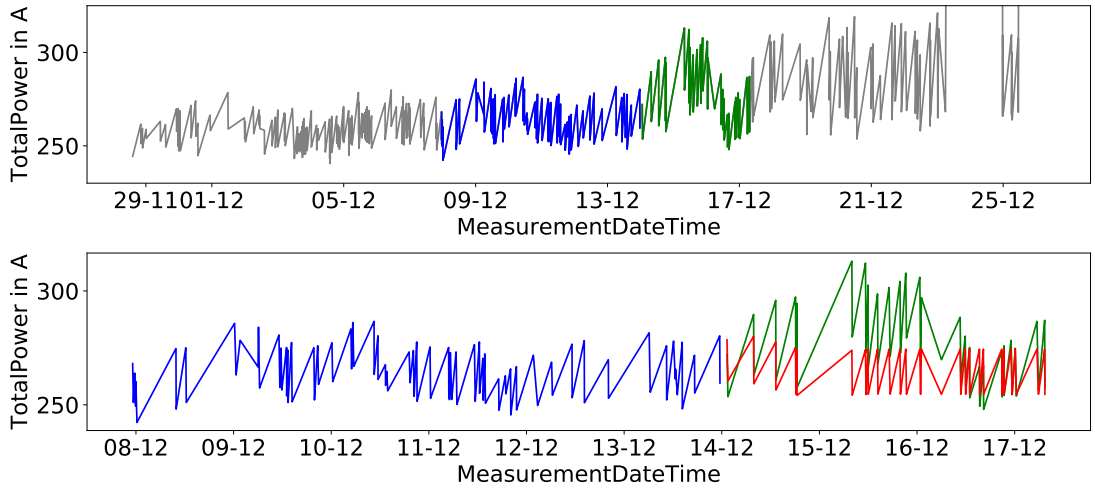

Figure 4.9: Shifted second training set (blue) and the shifted test set (green). At the top the data we do not consider (grey). The ARIMA(3,0,4) forecast (red) at the bottom

It can be seen that the ARIMA forecast behaves stiffly and goes straight out for each direction after a few time points.
Now we analyse the 3 data sets where the maintenance happened. We proceed as before. However, we now forecast up to 25 steps after the day the maintenance happened to see if the maintenance has an impact on the 50-step forecast of the test sets. Using the Dickey-Fuller test we conclude that the 3rd training set is stationary after being differenced one time. In the sorted list of the AICs, the first tuple that contains $d = 1$ is (4,1,2) with the AIC value 675.378678. We also perform a forecast based on the parameters of the best AIC (668.385923;(1,0,3)). The MSEs of the forecasts ((4,1,2)/(1,0,3)) are the following:

```
Mean squared error of the 1-step forecast:  26.5/11.7
Mean squared error of the 10-step forecast:  28.3/26.8
Mean squared error of the 50-step forecast:  136.1/107.2.
```

The 2 forecasts and their MSE are close to each other. They, as well as the day of the maintenance can be seen in Figure 4.10 where the 50-step forecast with the parameters (4,1,2) is almost overlapping the forecast with the parameters (1,0,3).

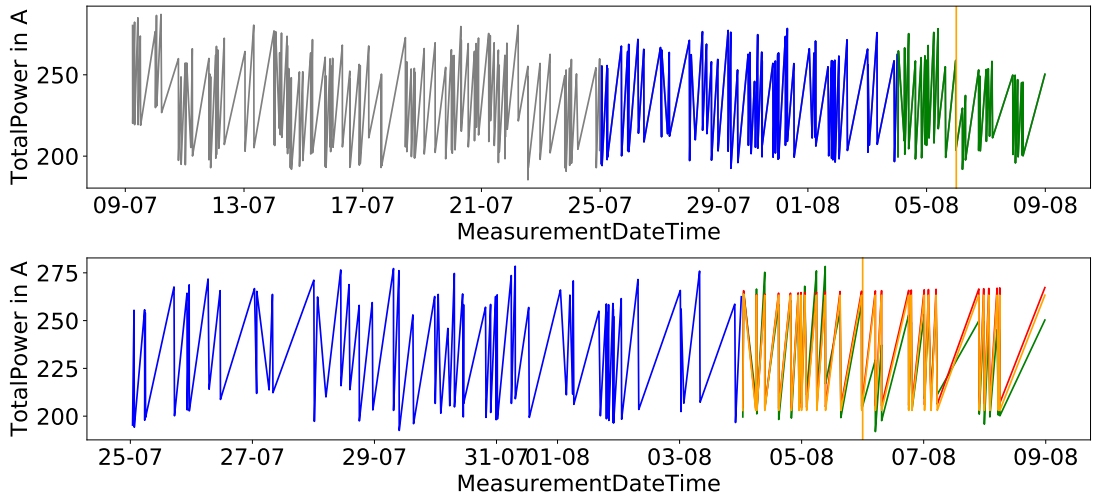

Figure 4.10: Third data set. Training set (blue), test set (green) and the day of the maintenance (orange vertical line). At the top the data we do not consider (grey). The ARIMA(4,1,2) forecast (red) and the ARIMA(1,0,3) forecast (orange) at the bottom

We are able to reject the null hypothesis and conclude stationarity for the 4th training set. Now we have a similar situation as before. The first parameter of the list of the best AICs and their respective parameter where $d = 0$ is (2,0,0) with an AIC of 716.302591. The AIC in the first place is (2,1,1) with an AIC of 714.076558. As before the MSEs ((2,0,0)/(2,1,1)) are shown below and the forecasts can be seen in Figure 4.11:

```
Mean squared error of the 1-step forecast:  6.5/0.2
Mean squared error of the 10-step forecast:  35.6/37.1
Mean squared error of the 50-step forecast:  191.9/228.7.
```

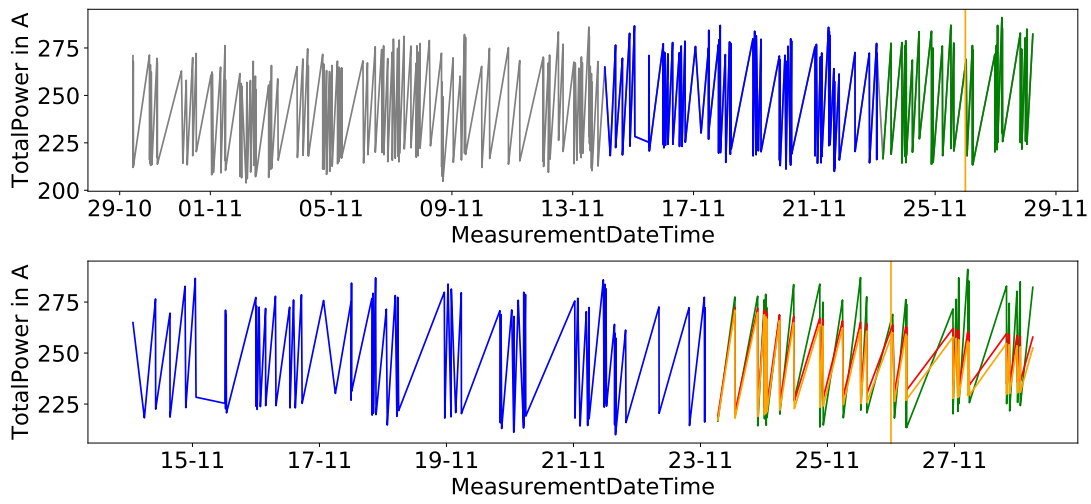

Figure 4.11: Fourth data set. Training set (blue), test set (green) and the day of the maintenance (orange vertical line). At the top the data we do not consider (grey). The ARIMA(1,0,3) forecast (red) and the ARIMA(2,1,1) forecast (orange) at the bottom

It is noticeable again that the 50-step forecasts perform almost identically. Unlike before the forecasts become closer to each other the more time passes. This should not happen. For the 5th training set, we concluded that the set is stationary after being differenced one time. Also, the top AIC is (1,1,4). The 50-step forecast can be seen in Figure 4.12 and the MSEs are the following:

```
Mean squared error of the 1-step forecast:  23.5
Mean squared error of the 10-step forecast:  41.0
Mean squared error of the 50-step forecast:  354.9.
```

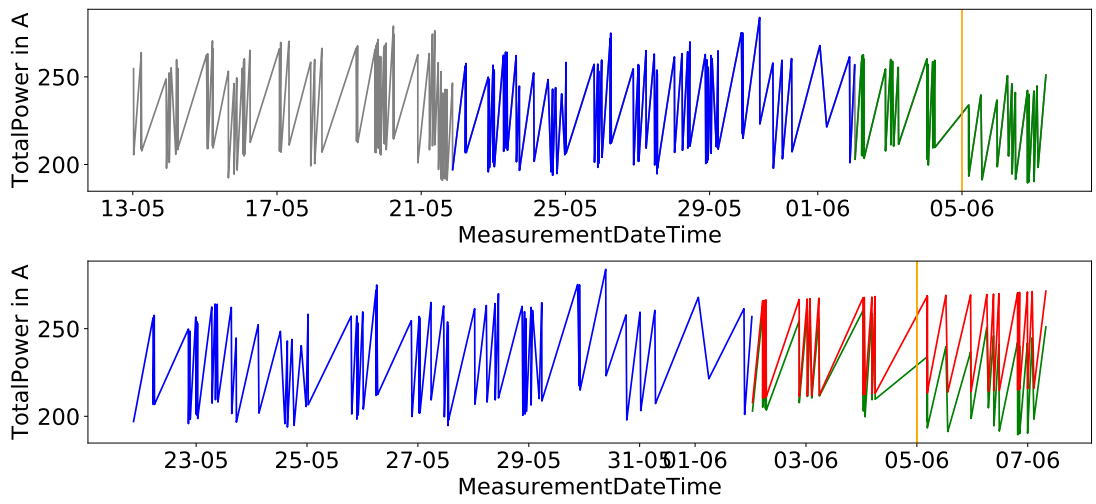

Figure 4.12: Fifth data set. Training set (blue), test set (green) and the day of the maintenance (orange vertical line). At the top the data we do not consider (grey). The ARIMA(1,1,4) forecast (red) at the bottom.

Now we want to use the VARMA model to fit a model and forecast the test sets. We split each training set of 100 data points into 50 of each direction. Again we use Statsmodels and the implemented VARMAX package to fit and forecast a model and use grid search with values from 0 to 4 to find parameters for p and q based on the best AIC. We start with the first data set. The output of the 3 best AICs and their given parameters for the first data set are shown below:

```
3 smallest AICs:
AIC              Parameter
545.421651       (1, 0)
545.428625       (2, 0)
549.292198       (3, 0).
```

We will now fit the model based on a VAR(1) process and get amongst other things the following output:

	coef	std err	z	P> |z|	[0.025	0.975]
intercept	52.1464	40.479	1.288	0.198	-27.191	131.484
L1.y1	0.6031	0.171	3.519	0.000	0.267	0.939
L1.y2	0.2144	0.259	0.828	0.408	-0.293	0.722
	coef	**std err**	**z**	**P> |z|**	**[0.025**	**0.975]**
intercept	52.2374	26.621	1.962	0.050	0.061	104.414
L1.y1	0.1240	0.125	0.994	0.320	-0.120	0.368
L1.y2	0.6363	0.180	3.541	0.000	0.284	0.989
	coef	**std err**	**z**	**P> |z|**	**[0.025**	**0.975]**
sqrt.var.y1	4.5749	0.586	7.805	0.000	3.426	5.724
sqrt.cov.y1.y2	1.7044	0.441	3.867	0.000	0.841	2.568
sqrt.var.y2	2.4746	0.241	10.249	0.000	2.001	2.948

Table 4.3: Part of the summary of the fitted VAR(1) model of the first training set using Statsmodels

That means the estimated VAR(1) process has the following form:

$$X_t = \begin{pmatrix} 52.1464 \\ 52.2374 \end{pmatrix} + \begin{pmatrix} 0.6031 & 0.2144 \\ 0.1240 & 0.6363 \end{pmatrix} X_{t-1} + Z_t,$$

where Z_t is $\mathcal{N}(0, \Sigma_z)$ distributed with $\Sigma_z = \begin{pmatrix} 4.5749 & 1.7044 \\ 1.7044 & 2.4746 \end{pmatrix}$.

When performing vector-valued forecasts or when we split the time series in two based on the respective measurement direction and perform forecasts for each direction, we will plot the two different time series separately (for example as at the top of Figure 4.13). Solid lines always stand for the measurement direction 0 whereas the dashed lines can be assigned to the measurement direction 1. To compare them with the ARIMA forecasts we zip the 2 directions up and calculate the MSE (illustrated at the bottom of Figure 4.13). Except for the shifted second test set the VARMA models performed better in terms of the MSE (Table 4.4, the top MSEs are in bold).

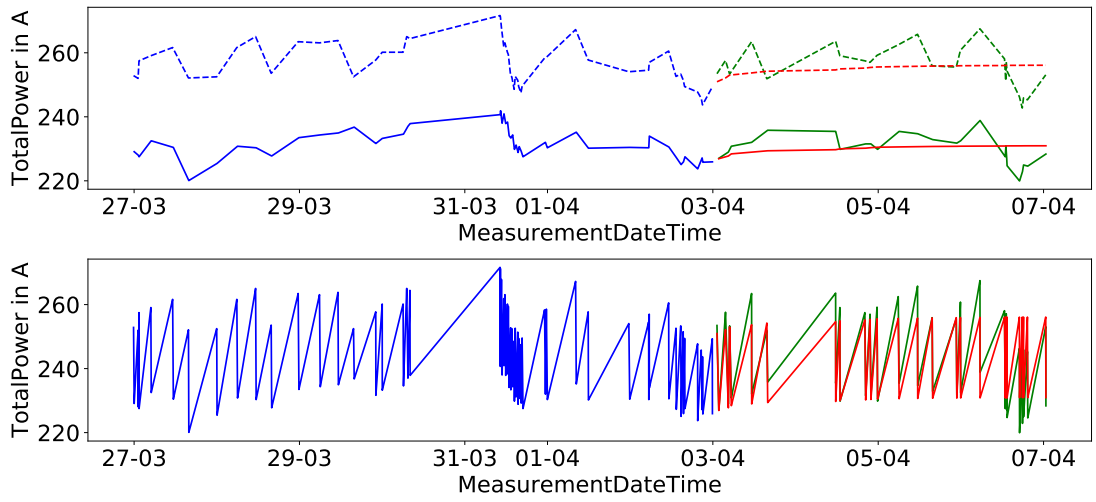

Figure 4.13: Training set (blue), test set (green) and the 50-step VAR(1) forecast (red) of the first test set

Method \ Test set		1	2 shifted	3	4	5
1-step	ARIMA	(2,1,2) 37.3	(3,0,4) 41.2	(4,2,1) 26.5 (1,0,3) 11.7	(2,0,0) 6.5 (2,1,1) 0.2	(1,1,4) 23.5
	VARMA	**(1,0) 6.4**	**(0,2) 21.6**	**(1,0) 1.2**	**(1,0) 0.1**	**(1,0) 8.4**
10-step	ARIMA	(2,1,2) 66.4	**(3,0,4) 154.5**	(4,2,1) 28.3 (1,0,3) 26.8	(2,0,0) 35.6 (2,1,1) 37.1	(1,1,4) 41
	VARMA	**(1,0) 19.5**	(0,2) 177.5	**(1,0) 26.1**	**(1,0) 9.1**	**(1,0) 9**
50-step	ARIMA	(2,1,2) 90.2	**(3,0,4) 278.8**	(4,2,1) 136.1 (1,0,3) 107.2	(2,0,0) 191.9 (2,1,1) 228.7	(1,1,4) 354.9
	VARMA	**(1,0) 32.6**	(0,2) 285.4	**(1,0) 106.9**	**(1,0) 34.7**	**(1,0) 127.9**

Table 4.4: MSEs of the ARIMA/VARMA forecasts.

4.1.2 Forecasts with exogenous variables

As already mentioned and discussed in [1, pp. 89-93], temperature and humidity might have an impact on the total power of the given railway switch. The relation between temperature and total power can be seen in Figures 4.14 and 4.15. We can observe the natural cycle of seasons in Figure 4.14. However, one season contains many data points and can vary from year to year which is why we refrain from using a seasonal model such as SARIMA. We want to see if we can improve the model through the consideration of the temperature at given time points. Logically, the farther away we consider the temperature (in the sense of time) of a fixed switch movement, the less influence it has on the target variable TotalPower of this fixed movement. We want to consider the actual temperature during a switch movement. With this approach, the VARMAX model of Definition 10 simplifies to

$$X_t = \nu + A_1 X_{t-1} + \ldots + A_p X_{t-p} + Z_t + B_1 Z_{t-1} + \ldots + B_q Z_{t-q} + M_0 U_t. \quad (4.1)$$

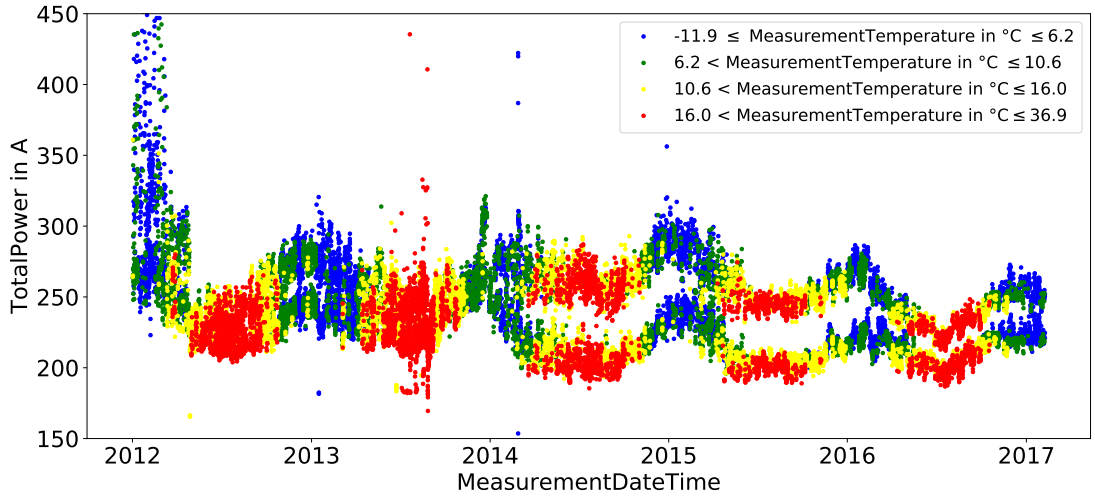

Figure 4.14: Progress of TotalPower, colourised by temperature

In this case, the parameter b of the general VARMAX equation (2.6) is always equal to 0 and therefore we will write the VARMAX$(p, q, 0)$ model as VARMAX(p, q) model. In this case the vector ν of constants consists of two values, $A_1, \ldots A_p, B_1, \ldots B_q$ and M_0 are 2×2 matrices. The external time series $(U_t)_{t \in T}$ is the 2-dimensional vector that stands for the temperature of each direction. Each row of the matrix M_0

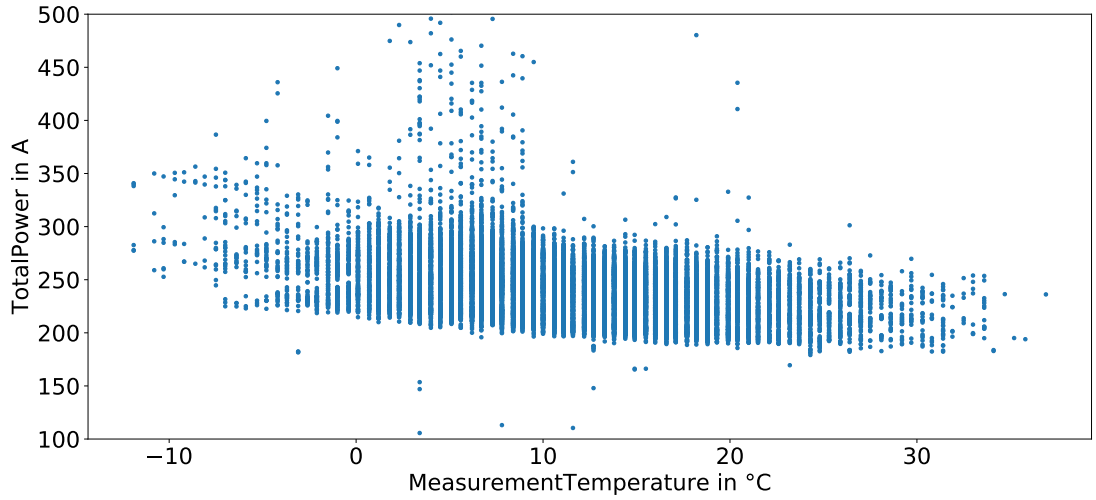

Figure 4.15: Temperature to TotalPower (zoomed in, some outliers cannot be seen)

can be interpreted as the weights we grant custody of the measured temperatures of the individual measurement directions. We explain the idea and the problems that arise using the VARMA(X) model in view of the data basis with an illustration of a VARX(1) model (a similar situation arises with higher orders, only autoregressive terms and with moving average terms) with Figure 4.16.

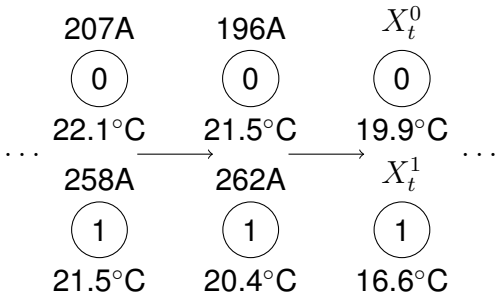

Figure 4.16: Illustration of a VARX(1) model

The number inside a circle represents the measurement direction. At the top of the circles, the values of the total power (here in amperes) and at the bottom the temperature values of each switch movement can be seen. Both directions are summarised in a vector (the order depends on which direction we are starting). Each time step of a VARX(1) model is represented by an arrow to the right. The missing values of the total power over the right circles ($X_t^0,\ X_t^1$) of the VARX(1)

process are computed as follows:

$$\begin{pmatrix} X_t^0 \\ X_t^1 \end{pmatrix} = \begin{pmatrix} \nu_1 \\ \nu_2 \end{pmatrix} + \begin{pmatrix} a_{11}196 + m_{12}262 + m_{11}19.9 + m_{12}16.6 \\ a_{21}196 + a_{22}262 + m_{21}19.9 + m_{22}16.6 \end{pmatrix} + \begin{pmatrix} Z_t^0 \\ Z_t^1 \end{pmatrix},$$

where ν_1, ν_2 are constants, $(a_{ij})_{1\leq i,j\leq 2}$ are the components of the matrix A_1 and $(m_{ij})_{1\leq i,j\leq 2}$ are the components of the matrix M_0 of (4.1). The entries Z_t^0 and Z_t^1 belong to the vector-valued white noise of Z_t of (4.1). We see that the last two known values of the total power of each direction are used to compute X_t^0 and X_t^1 as well as the actual measured temperature of each direction. It can be advantageous when computing the total power of a measurement direction to also use the measurement temperature and total power value of the other direction, however if we want to compute for example X_t^0, and a long time interval lies between the two measurements 0 and 1, the model might use a temperature or the total power value of the measurement direction 1 that is not influencing X_t^0 and even worse it can distort the model. The same applies for X_t^1, it may be that it uses the outdated value which got measured for the other direction and is no longer relevant. We discover the parameters for p and q of the VARMAX model as before in the VARMA case using the AIC. The output is similar to a VARMA model, in addition, we get the values of the Matrix M_0 as an output, shown in Table 4.5.

	coef	**std err**	**z**	**P$>$ |z|**	**[0.025**	**0.975]**
beta.x1	-1.4444	1.312	-1.101	0.271	-4.017	1.128
beta.x2	0.4156	1.193	0.348	0.727	-1.922	2.753
beta.x1	-0.2613	1.029	-0.254	0.799	-2.278	1.755
beta.x2	-0.2778	1.077	-0.258	0.796	-2.388	1.833

Table 4.5: Part of the summary using Statsmodels. Coefficients of the matrix M_0 of the estimated VARMAX(1,3) model of the first training set

It follows that the estimated matrix has the form $M_0 = \begin{pmatrix} -1.4444 & 0.4156 \\ -0.2613 & -0.2778 \end{pmatrix}$. We also want to analyse the total power of a measurement direction independently from the other direction. We divide the training set into 2 training sets consisting of 50 data points for each direction. In Definition 8 we considered an ARMA model and additionally the last b inputs of an external series. Now we do not want to consider the last b inputs of an external series but the actual input of the external series.

In this case the measured temperature during a switch movement. Modifying the ARMAX equation for this purpose we end up with the equation

$$X_t = c + \sum_{i=1}^{p} \phi_i X_{t-i} + Z_t + \sum_{j=1}^{q} \theta_j Z_{t-j} + \eta U_t, \tag{4.2}$$

where η is the scalar for the univariate external series U_t. The equation (4.2) has the same form as a linear regression with ARMA errors we considered in Definition 12 with a constant $c \in \mathbb{R}$ when choosing $k = 1$, i.e. we use 1 regressor. To see if it is beneficial to model the error term of a linear regression with an ARMA model[2], we will also forecast the respective test set with a usual linear regression model using temperature as regressor[3]. To forecast with a linear regression model, we divide each training set of 100 data points into 2 training sets of 50 data points for each direction. Then we estimate the intercept and slope of the respective regression line (minimisation of the residual sum of squares). Based on the estimated regression line we forecast the future values of TotalPower (c.f. Figure 4.18). We compute a maximum of 25-step forecasts based on the temperature of the given time for each direction. After computing 25 step forecasts for each direction (Figure 4.18) we zip

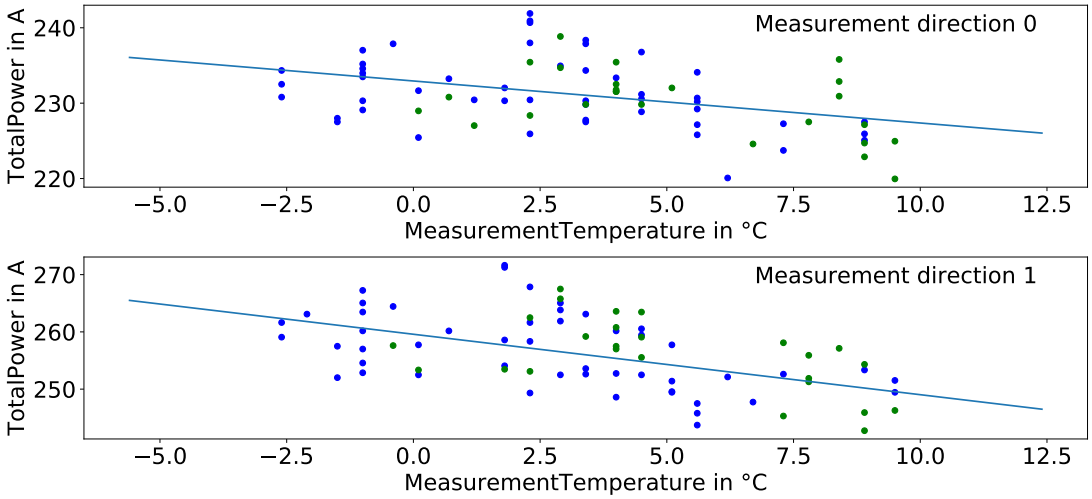

Figure 4.17: The 50 data points of the training set (blue), the estimated regression line and the future 25 values (green) of the first test set

them together to get a 50-step forecast and calculate the respective MSEs.

[2]Using tsa.arima_ model.ARMA of Statsmodels
[3]Using linear_ model.LinearRegression of sklearn

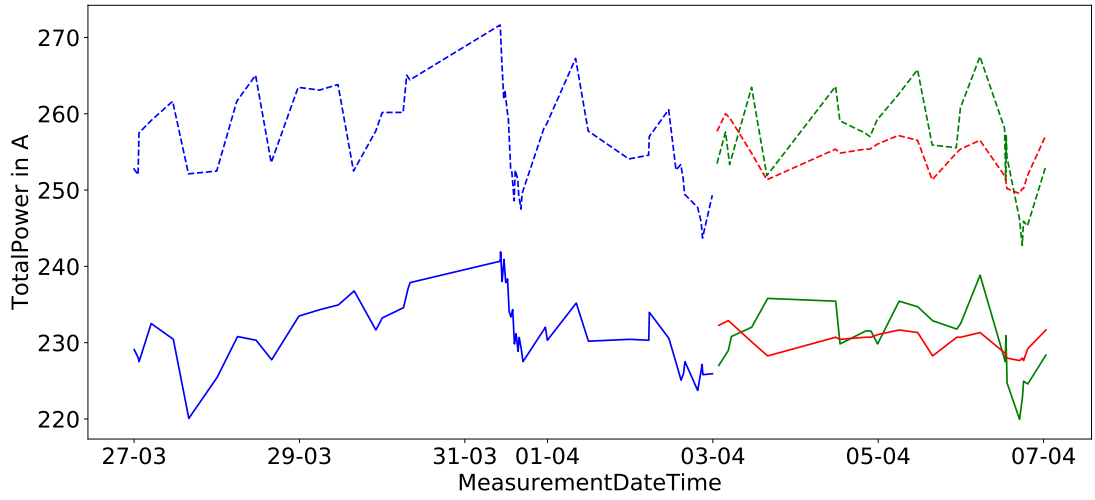

Figure 4.18: Training set (blue), test set (green) and the linear regression forecasts (red) of the respective measurement direction of the first test set

4.1.3 Forecasts of each data set

We want to apply the discussed forecast methods for each data set. We observe in Table 4.6 that the linear regression forecast that we already saw in Figure 4.18 was the best 50-step forecast among all of the forecasting methods. Using the linear

Test set 1	1-step	10-step	50-step
ARIMA(2,1,2)	37.3	66.4	90.2
VARMA(1,0)	6.4	**19.5**	32.6
VARMAX(1,3)	23.8	27.6	97.3
Linear regression with ARMA errors 0-(2,2), 1-(1,0)	**2.2**	27.4	27.7
Linear regression	17.8	24.2	**23**

Table 4.6: MSEs of the forecasts of the first test set

regression with ARMA errors model we can improve the linear regression forecast for the 1-step case. The MSEs do not get very big over time and it seems that the error happened suddenly because the forecasts do not deviate much from the test set. Reconsidering the shifted training and test set of the second data set, we see in Table 4.7 that the MSEs of each test set for 50-step forecasts got very big compared to the 50-step forecasts of the first test set.

Test set 2 shifted	1-step	10-step	50-step
ARIMA(3,0,4)	41.2	154.5	278.8
VARMA(0,2)	21.6	177.5	285.4
VARMAX(3,0)	41.2	**146.3**	**256.5**
Linear regression with ARMA errors 0-(1,0), 1-(3,0)	8.5	167.4	257.3
Linear regression	**3.2**	173.2	263.8

Table 4.7: MSEs of the forecasts of the shifted second test set

We can observe in Figure 4.19, that TotalPower does not depend much on the measured temperature in the test set.

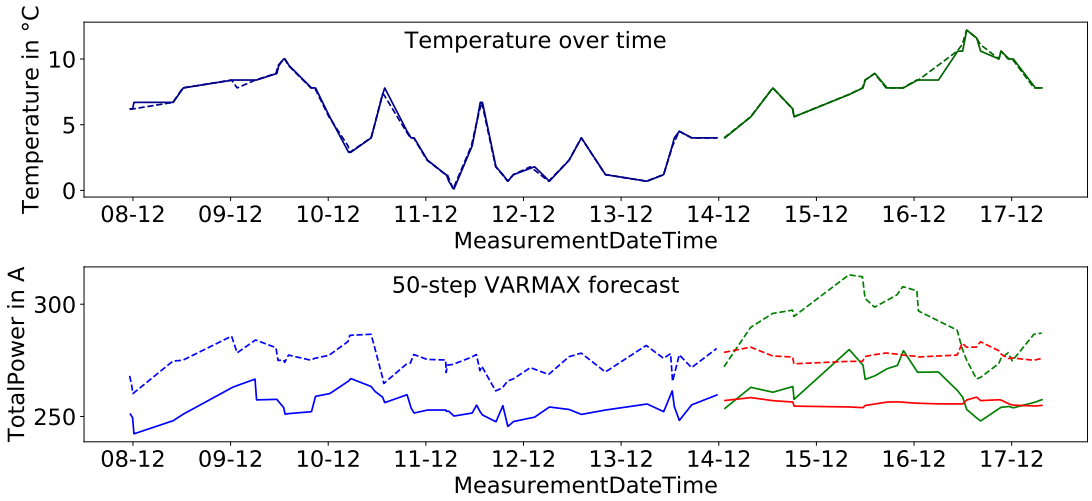

Figure 4.19: Temperature change of the shifted second training and test set (top) and the corresponding training set (blue), test set (green) and the 50-step VARX(3) forecast (red) at the bottom

The ups and downs of TotalPower in the test set were not caused by a large variation in temperature. Considering the slowly rising temperature of the test set we would expect that TotalPower of the test set would slowly decay and not spike up and down. Neither the VARMAX forecast at the bottom of Figure 4.19 nor the other 2 forecasts that contain the temperature as exogenous variable (Figure 4.20) forecast this spike. We conclude that the ups and downs of TotalPower were not caused by temperature and the railway switch was not behaving normally.

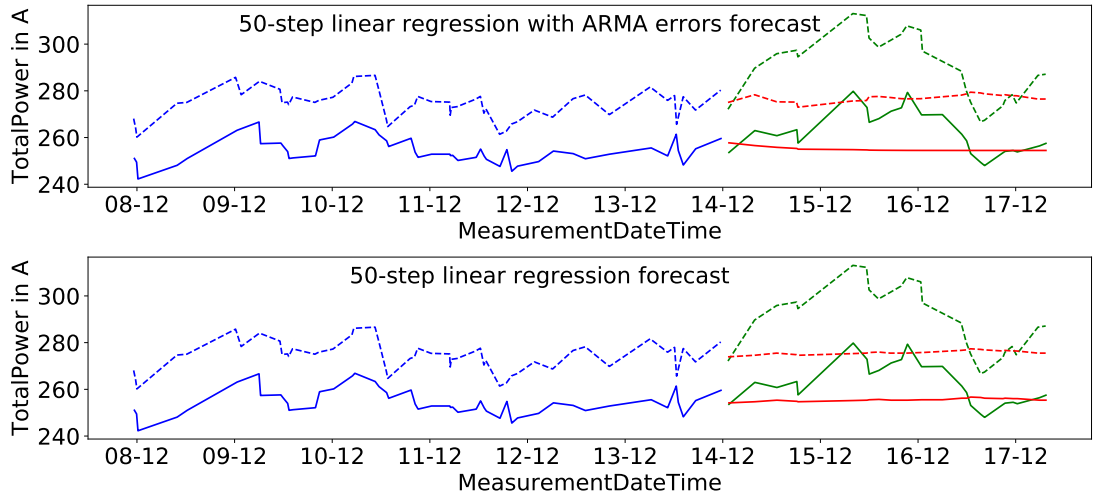

Figure 4.20: Training set (blue), test set (green) and the other forecasts that consider the measured temperature (red) of the shifted second test set

Looking at the MSEs and the linear regression forecast of the 3rd test set (Table 4.8 and Figure 4.21), we can observe that the maintenance affected the total power consumption of the railway switch for the measurement direction 1 (dashed green line). Before the day the maintenance happened the model was describing the energy consumption for the measurement direction 1 of the switch very well but not afterwards. On the contrary, looking at the MSEs of the 50-step forecasts of the 4th test set (Table 4.9), we see that they are much smaller compared to the 50-step forecast MSEs of the 3rd test set because the maintenance action was only inspection and the model is describing the data well.

Test set 3	1-step	10-step	50-step
ARIMA(4,2,1)	26.5	28.3	136.1
ARIMA(1,0,3)	11.7	26.8	107.2
VARMA(1,0)	**1.2**	26.1	**106.9**
VARMAX(1,0)	**1.2**	**17.8**	133.4
Linear regression with ARMA errors 0-(2,3), 1-(0,0)	5.5	17.9	132.5
Linear regression	16	20.7	131.6

Table 4.8: MSEs of the forecasts of the 3rd test set

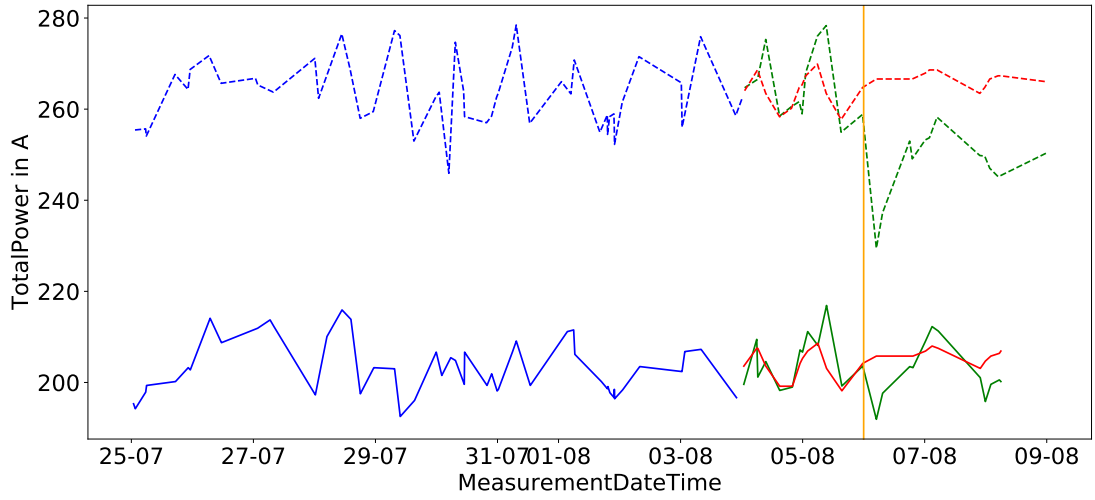

Figure 4.21: Training set (blue), test set (green), the linear regression forecast of the 3rd test set (red) and the day the maintenance took place (orange vertical line)

Test set 4	1-step	10-step	50-step
ARIMA(2,0,0)	6.5	35.6	191.9
ARIMA(2,1,1)	0.2	37.1	228.7
VARMA(1,0)	**0.1**	**9.1**	**24.7**
VARMAX(1,0)	17.5	12.8	38.3
Linear regression with ARMA errors 0-(1,0), 1-(0,3)	0.3	16.6	36.4
Linear regression	19.9	20.9	39.8

Table 4.9: MSEs of the forecasts of the 4th test set

The MSEs of the 50-step forecasts of the 5th test set (c.f. Table 4.10) lie between the last two. Among the 50-step forecasts, the predictions right before and right after the maintenance took place were not as good as the other ones that are farther away from the day of the maintenance (Figure 4.22). It seems that the switch performed normally as before shortly after the maintenance.

Test set 5	1-step	10-step	50-step
ARIMA(1,1,4)	23.5	41	354.9
VARMA(1,0)	8.4	9	127.9
VARMAX(1,0)	**0**	5.4	**59.5**
Linear regression with ARMA errors 0-(0,3), 1-(0,2)	2.8	**2.1**	69.7
Linear regression	3.6	4	71.2

Table 4.10: MSEs of the forecasts of the 5th test set

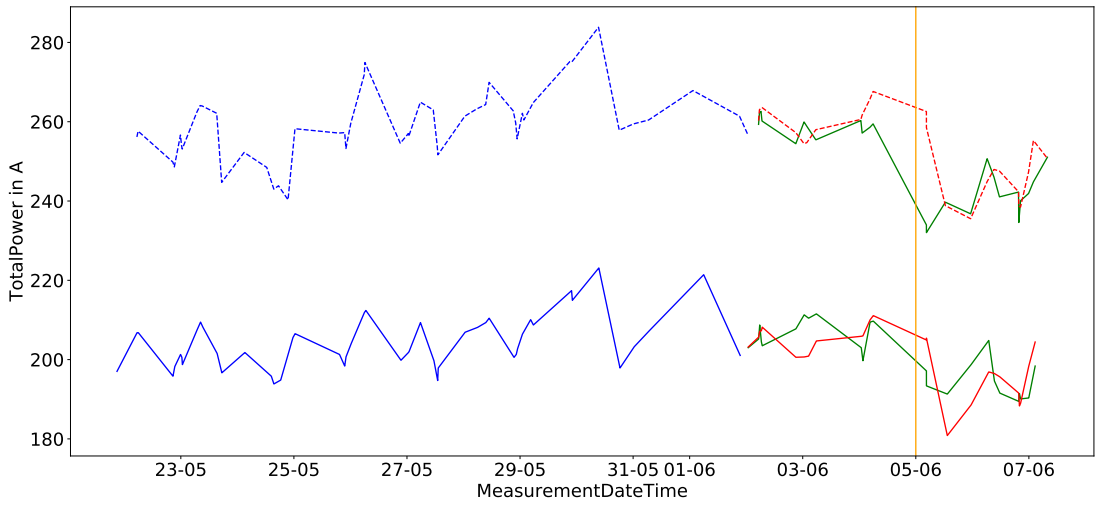

Figure 4.22: Training set (blue), test set (green), the VARX(1) forecast of the 5th test set (red) and the day the maintenance took place (orange vertical line)

4.2 Switch 2

The data of Switch 2 consists of 66797 observations recorded from 01-01-2012 to 20-03-2018 and 9 variables (MeasurementID, MeasurementDirection, MeasurementTemperature, MeasurementDateTime, NumberOfEngines, Engine, DataStart, DataEnd, Data). Unlike in Switch 1, number of engines is equal to 2. Each engine belongs to a measurement direction. To get an approximation of the total power as a derived feature parameter, we sum up the values in "Data" (c.f. Figure 4.23) and generate a new variable "TotalPower". Similar to Switch 1, TotalPower is also

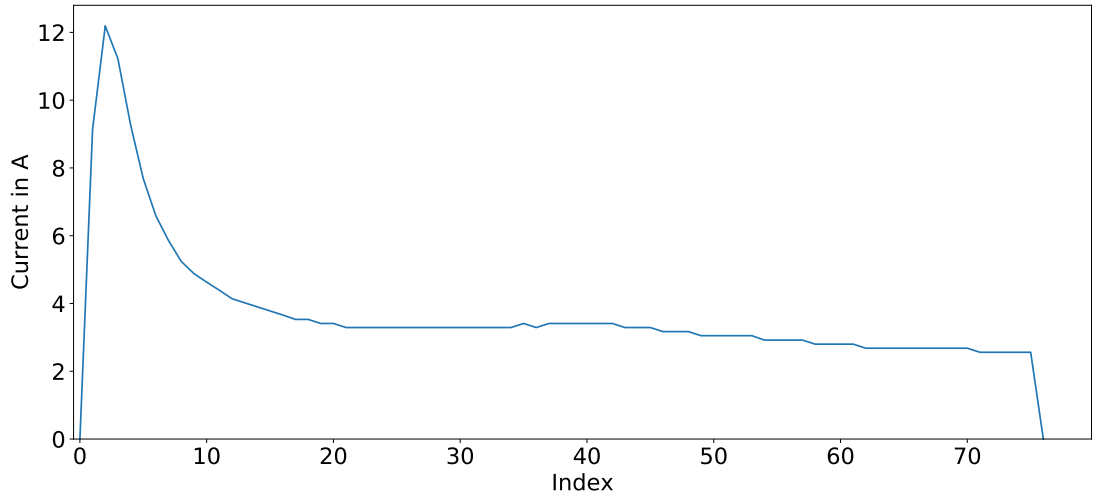

Figure 4.23: A plot of the data points of one expression of the variable Data

dependent on the measurement direction (Figure 4.24) and strongly dependent on the measured temperature (Figure 4.25).

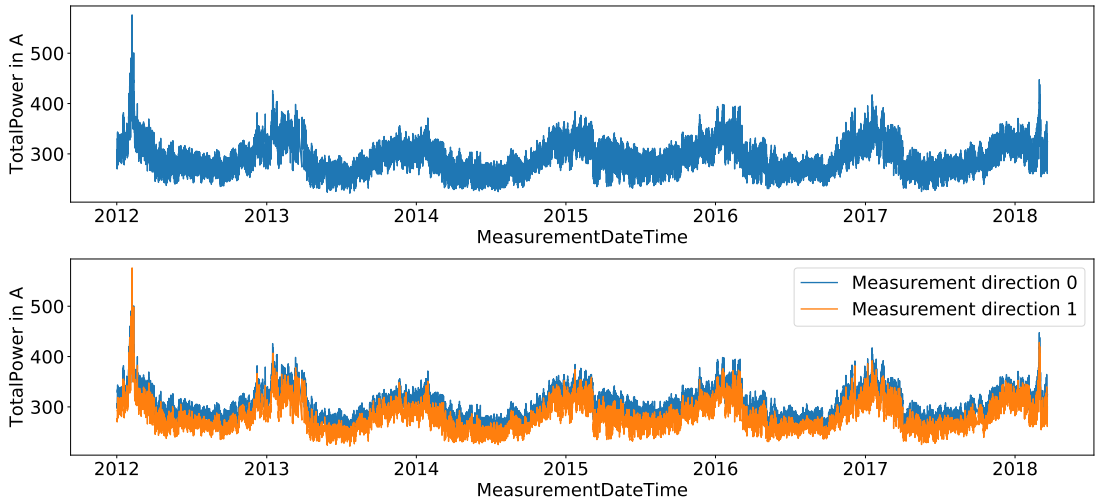

Figure 4.24: The constructed time series TotalPower (top), split in the two measurement directions (bottom)

Only 1 failure has been recorded at 29-01-2014 and 88 maintenance actions were listed. We also have 143 time points reported where the switch was heated. In this chapter, we want to test the different forecasting methods against each other. We pick 10 sets where no failure, maintenance and heating occurred. We proceed as in

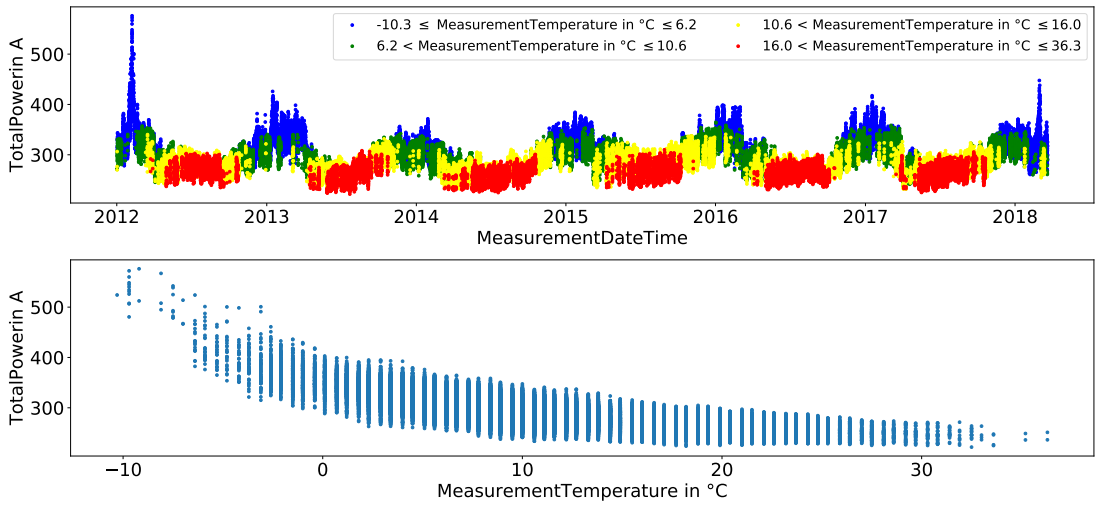

Figure 4.25: TotalPower, colourised with the associated temperature (top) and MeasurementTemperature to TotalPower (bottom)

Switch 1 except we end the test set 100 data points before maintenance has taken place or heating was performed to get rid of a possible bias that could occur before maintenance or heating. The data sets end respectively 100 datapoints before an event (maintenance or heating) and go back until the closest event (maintenance or heating) before. From these sets we only use the last 150 data points (100 data points for training the model and maximum 50 for forecasting), they can be seen in Tables 4.11, 4.12 and are colourised in Figure 4.26.

Data set	1	2	3	4	5
Time frame	from 10-06-2012 19:17:00 to 01-08-2012 18:10:00	from 07-06-2013 06:29:00 to 24-07-2013 08:29:00	from 15-09-2013 21:24:00 to 15-10-2013 23:02:00	from 24-03-2014 18:26:00 to 15-05-2014 05:42:00	from 13-09-2014 18:51:00 to 13-10-2014 09:17:00
Number of data points	1335	1113	864	1632	960

Table 4.11: First 5 data sets of Switch 2

Data set	6	7	8	9	10
Time frame	from 01-04-2015 11:52:00 to 06-05-2015 22:41:00	from 21-06-2015 06:11:00 to 03-09-2015 01:11:00	from 21-03-2016 13:23:00 to 20-07-2016 12:08:00	from 18-09-2016 11:11:00 to 04-10-2016 00:59:00	from 20-05-2017 20:34:00 to 07-09-2017 00:53:00
Number of data points	1035	2074	3664	462	3367

Table 4.12: Last 5 data sets of Switch 2

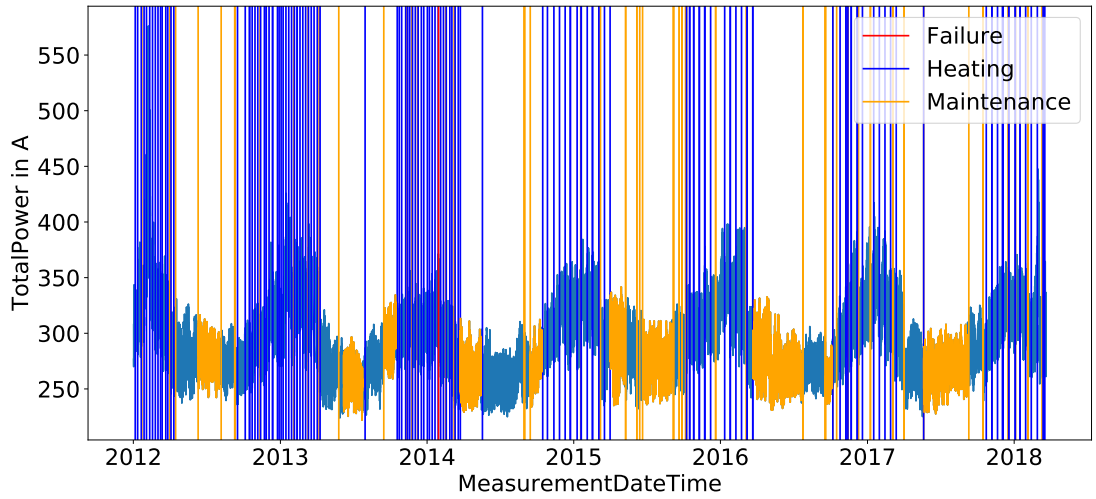

Figure 4.26: Failure (red vertical line), maintenance (orange vertical lines), heating of the switch (blue vertical lines) and the 10 data sets we want to analyse, colourised in orange

4.2.1 1-step forecasts

We are starting to compare the 1-step forecasts with the discussed forecasting methods shown in Tables 4.13 and 4.14. Except for the 2nd and 3rd training set the method for defining the parameter d for the ARIMA(p, d, q) model coincided with the parameter d of the best found AIC. For the 2nd training set, we rejected the null hypothesis and concluded that the training set is stationary. However, in the list of the smallest AICs the best parameters found were (4,1,3). For the 3rd training set, we concluded that the training set is stationary after we differenced it

one time. In this case, the best parameters based on the smallest AIC were (2,0,1). We included both parameters and their MSEs of the forecasts in those two cases. For the 1-step forecasts, it was not beneficial to include the external variable of the measured temperature. The MSEs calculated with the forecasts without the exogenous variable are even better in most cases. Unlike in Switch 1, the forecasts of the VARMA model are not better than the forecasts with the ARIMA models which is probably caused by the gap between the measurement direction (i.e. the difference between the (discrete) TotalPower of the two measurement directions) which is not as big as in Switch 1, although the parameters can be reduced. In 18 out of 20 cases, we estimated a VAR(1) or VARX(1) as the best model to use in the VARMA/VARMAX case.

Method \ Test set	1	2	3	4	5
ARIMA	**(4,0,3)** **0**	(4,0,2) 15.3 (4,1,3) 15.6	**(2,1,1)** **13.6** (2,0,1) 41.5	(2,1,3) 3.3	**(2,1,3)** **3.8**
VARMA	(1,0) 5.1	**(1,0)** **5.2**	(1,0) 34.9	(1,0) 8.4	(1,0) 4.4
VARMAX	**(1,0)** **0**	(1,0) 14.6	(1,0) 29.6	**(1,0)** **0.1**	(1,0) 10.8
Linear regression with ARMA erros	0-(4,3) 1-(0,2) 0.1	0-(0,1) 1-(1,0) 184.6	0-(0,3) 1-(1,0) 61.2	0-(4,2) 1-(1,0) 0.7	0-(0,3) 1-(0,0) 6.7
Linear regression	4.1	165.3	60.6	31.3	6.7

Table 4.13: MSEs of the 1-step forecasts of the first 5 test sets

Method \ Test set	6	7	8	9	10
ARIMA	(3,1,2) 2.1	**(2,1,1)** **71.5**	**(1,1,3)** **0.5**	(4,1,0) 26.9	(1,1,1) 1.2
VARMA	**(1,0)** **1.4**	(1,0) 103.3	(1,0) 1.4	**(2,0)** **18.6**	(1,0) 1
VARMAX	(1,1) 6	(1,0) 138.7	(1,0) 4.3	(1,0) 19.5	**(1,0)** **0.4**
Linear regression with ARMA erros	0-(0,4) 1-(1,0) 1.5	0-(1,0) 1-(0,3) 107.4	**0-(4,1)** **1-(0,4)** **0.5**	0-(1,1) 1-(2,2) 80.6	0-(3,2) 1-(2,3) 2.6
Linear regression	17	175.2	8.2	48.5	19.7

Table 4.14: MSEs of the 1-step forecasts of the last 5 test sets

4.2.2 10-step forecasts

The situation changes observing the tables of the 10-step forecasts (Tables 4.15 and 4.16). The 10-step forecasts based on an ARIMA model do not perform as well as in the 1-step case.

Method \ Test set	1	2	3	4	5
ARIMA	(4,0,3) 17.3	(4,0,2) 269 (4,1,3) 192.5	(2,1,1) 49.3 (2,0,1) 37.6	(2,1,3) 118.3	(2,1,3) 91.3
VARMA	**(1,0)** **16.1**	(1,0) 293.5	**(1,0)** **35.1**	(1,0) 53.6	**(1,0)** **16.3**
VARMAX	(1,0) 44.2	**(1,0)** **46.6**	(1,0) 44.2	(1,0) 18	(1,0) 48.2
Linear regression with ARMA erros	0-(4,3) 1-(0,2) 65.2	0-(0,1) 1-(1,0) 80.3	0-(0,3) 1-(1,0) 40.5	**0-(4,2)** **1-(1,0)** **11.5**	0-(0,3) 1-(0,0) 49.4
Linear regression	43.2	75.8	53.4	31.3	43.5

Table 4.15: MSEs of the 10-step forecasts of the first 5 test sets

Method \ Test set	6	7	8	9	10
ARIMA	(3,1,2) 101.4	(2,1,1) 28.3	(1,1,3) 50.7	**(4,1,0)** **38**	(1,1,1) 12.8
VARMA	(1,0) 71.1	(1,0) 55.9	(1,0) 36	(2,0) 103.6	**(1,0)** **2.9**
VARMAX	(1,1) 88.5	(1,0) 20	(1,0) 19.2	(1,0) 34.9	(1,0) 32.8
Linear regression with ARMA erros	**0-(0,4)** **1-(1,0)** **60**	**0-(1,0)** **1-(0,3)** **15.5**	**0-(4,1)** **1-(0,4)** **18.6**	0-(1,1) 1-(2,2) 49.2	0-(3,2) 1-(2,3) 26.3
Linear regression	93.9	24.5	25	46.3	49.1

Table 4.16: MSEs of the 10-step forecasts of the last 5 test sets

4.2.3 50-step forecasts

In the tables of the 50-step forecasts (Tables 4.17 and 4.18), we see that the ARIMA and VARMA models perform very poorly compared to the models that contain exogenous variables.

Method \ Test set	1	2	3	4	5
ARIMA	(4,0,3) 105	(4,0,2) 155.2 (4,1,3) 95.6	(2,1,1) 72.4 (2,0,1) 33.1	(2,1,3) 126.8	(2,1,3) 285.9
VARMA	(1,0) 102.7	(1,0) 161.3	(1,0) 29.6	(1,0) 78.4	**(1,0)** **46.8**
VARMAX	**(1,0)** **35.5**	**(1,0)** **33.6**	(1,0) 28.5	(1,0) 69.4	(1,0) 62.3
Linear regression with ARMA erros	0-(4,3) 1-(0,2) 46.6	0-(0,1) 1-(1,0) 90.4	**0-(0,3)** **1-(1,0)** **27.7**	**0-(4,2)** **1-(1,0)** **32.7**	0-(0,3) 1-(0,0) 61.8
Linear regression	39.6	80.3	29.8	43.1	59.7

Table 4.17: MSEs of the 50-step forecasts of the first 5 test sets

Method \ Test set	6	7	8	9	10
ARIMA	(3,1,2) 362.7	(2,1,1) 77.1	(1,1,3) 140.9	(4,1,0) 90.2	(1,1,1) 269.1
VARMA	**(1,0)** **64.8**	(1,0) 127.6	(1,0) 112.5	(2,0) 121.1	(1,0) 69
VARMAX	(1,1) 98.8	(1,0) 45.8	(1,0) 69.5	**(1,0)** **31.3**	(1,0) 68.2
Linear regression with ARMA erros	0-(0,4) 1-(1,0) 94.7	**0-(1,0)** **1-(0,3)** **41.4**	0-(4,1) 1-(0,4) 55.1	0-(1,1) 1-(2,2) 41.4	**0-(3,2)** **1-(2,3)** **65.9**
Linear regression	100.5	41.7	**36.9**	45.8	113.4

Table 4.18: MSEs of the 50-step forecasts of the last 5 test sets

An ARIMA model should not be used for 50-step forecasts. The model can obviously not forecast the increase in total power due to a decrease in temperature. In the worst case, the 50-step forecast based on an ARIMA model gets completely out of hand and moves in a different direction as the test set, for example, such as in the last test set (Figure 4.27). The VARMA model prevents this extreme behaviour by converging to a constant for each direction (Figure 4.28) but it cannot be recommended for getting a precise forecast.

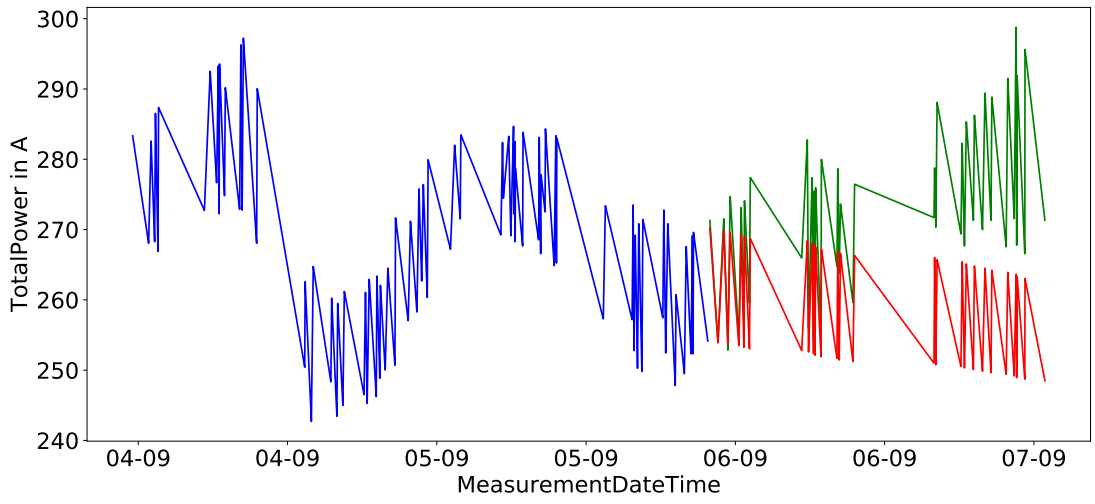

Figure 4.27: Training set (blue), test set (green) and the ARIMA(1,1,1) forecast of the last test set (red)

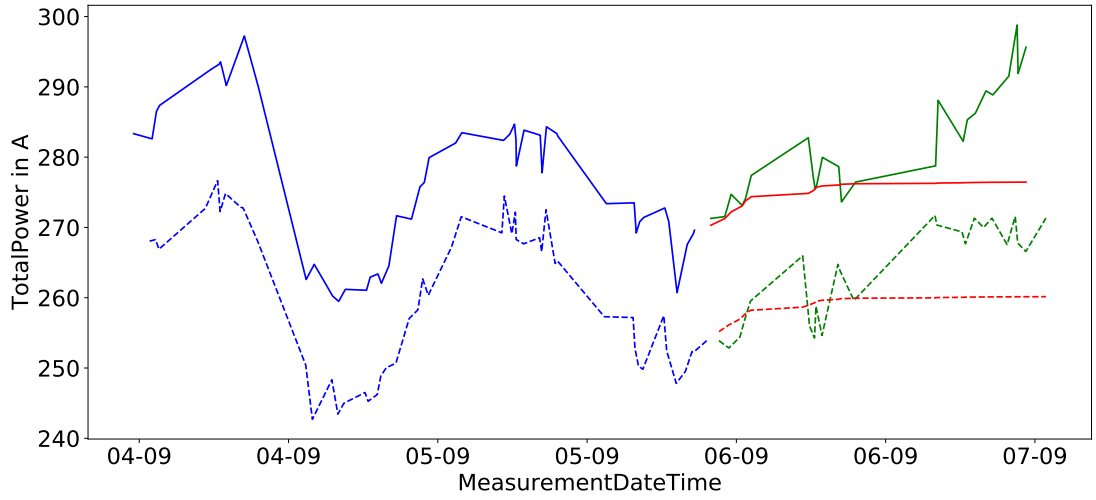

Figure 4.28: Training set (blue), test set (green) and the VAR(1) forecast of the last test set (red)

The more steps we want to forecast, the more advisable it is to use a method which includes exogenous variables. In 8 out of 10 cases for the 50-step forecasts, the models that considered the current measured temperature were superior to the models that did not. The VARMA forecasts only performed preferably for the 5th and 6th test set compared to the other models. We now want to investigate why this is the case. The situation is illustrated for the 5th test set in Figure 4.29. We can observe that the temperature does not change a lot. The difference between the lowest and highest measured temperature in this time frame is only 7.6℃, whereas in comparison the difference between the lowest and highest measured temperature recorded in the span of the 2nd training and test set (where the VARMAX model performed way better than the VARMA model) is 16.4℃ (c.f. Figure 4.30)

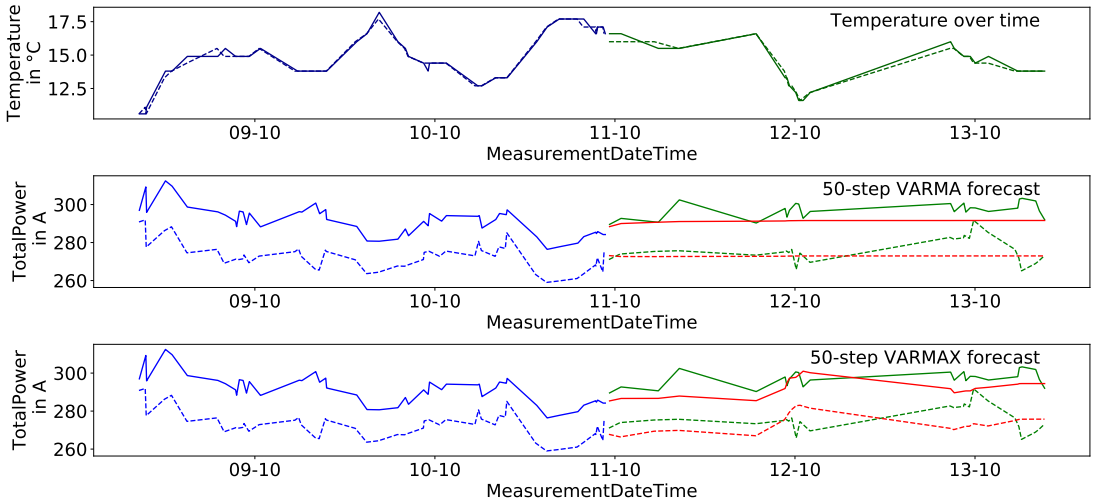

Figure 4.29: The progress of temperature of the 5th training and test set (top), training set (blue), test set (green) as well as the VAR(1) and VARX(1) forecasts (red) at the middle and bottom

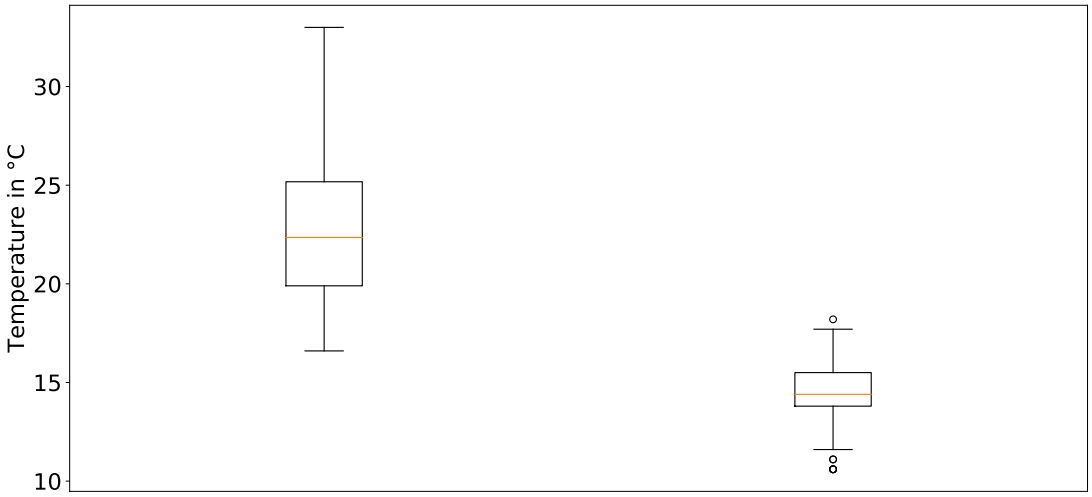

Figure 4.30: A boxplot of the temperature of the 2nd training and test set (left) and a boxplot of the temperature of the 5th training and test (right)

After creating the same three plots as before for the 2nd data set (Figure 4.31), we see that the temperature rises by almost 10 °C by day compared to the night and the VARMAX model can model these ups and downs of the temperature very well compared to the VARMA model which only converges to a constant for each direction. In

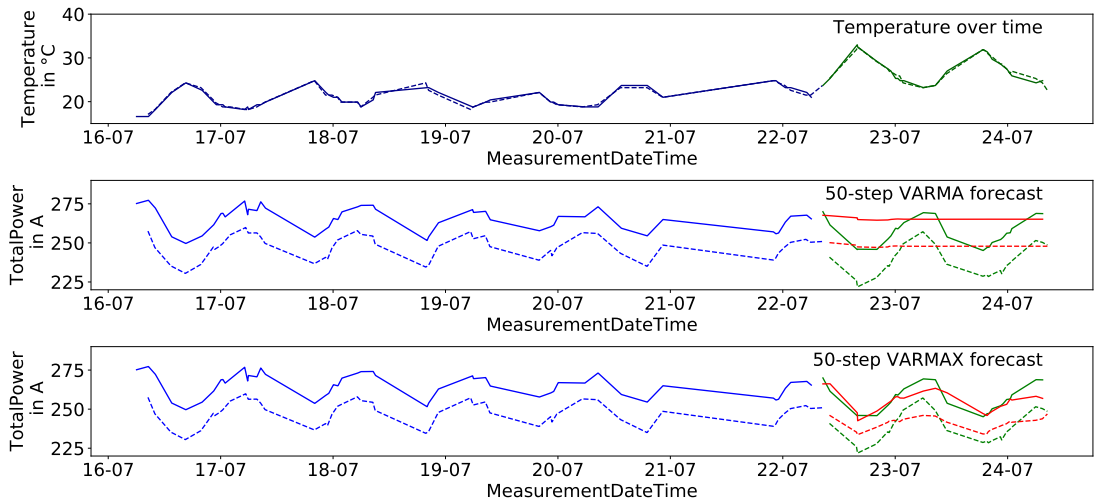

Figure 4.31: The progress of temperature of the 2nd training and test set (top), training set (blue), test set (green) as well as the VAR(1) and VARX(1) forecasts (red) at the middle and bottom

the time frame of the 5th training and test set, the measured temperature does not vary much and therefore also TotaPower does not vary much and a VARMA model can produce good forecasts. Also, it could be the case, as already discussed, that the VARMAX model has taken irrelevant data from the other direction into account, which leads to distortion. In the 6th data set, we can observe a peak in temperature of the test set at about 20.4℃ in the beginning (c.f. Figure 4.32). After the peak in temperature, the temperature does not vary much and we can conclude with the same argumentation as before for the 5th data set that TotalPower does not vary much and therefore the VARMA model can perform a good forecast. This peak is higher than the highest measured temperature in the training set (black horizontal line at the top) and it is therefore not surprising but reasonable that the forecast of TotalPower with the VARMAX model in this area is lower than the minimum value of TotalPower in the training set for the measurement direction 1 (black horizontal line at the bottom). It could be the case that the total power does not only depend on the measurement temperature and there are other variables we did not consider which also influenced the target variable TotalPower at that time. As mentioned before, these two forecasts of the 5th and 6th test set are exceptions and we clearly saw that the temperature influenced TotalPower and therefore a model including exogenous variables for a long forecasting horizon should be used.

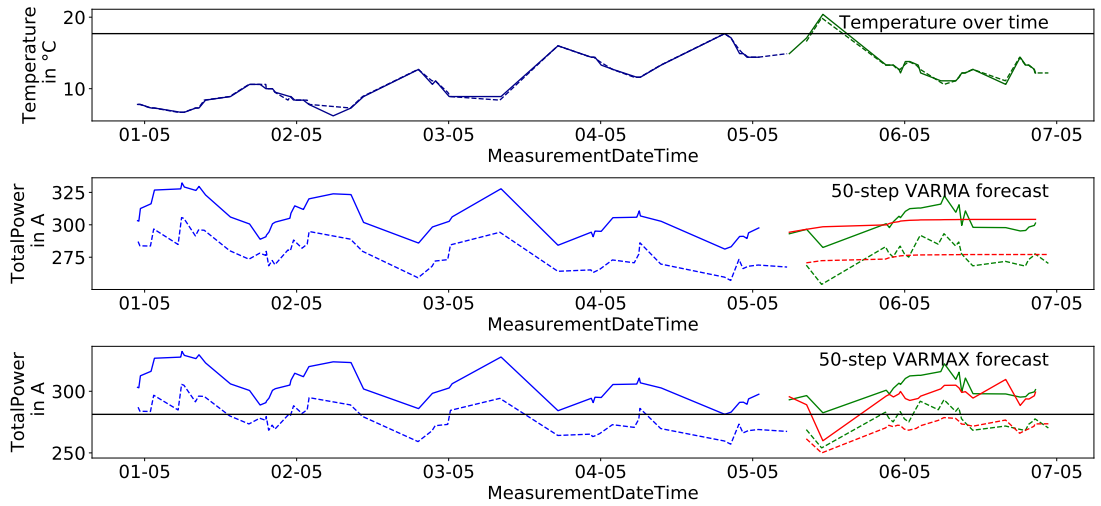

Figure 4.32: Progress of temperature of the 6th training and test set (top), training set (blue), test set (green) and the VAR(1) and VARMAX(1,1) forecasts (red) at the middle and bottom

The plot of the measured temperature to TotalPower in Figure 4.25 does not look linear. A good example of this non-linear behaviour is the last data set. The test set does not behave as the estimated linearity of the training set (left part of Figure 4.33). Therefore the exogenous forecasting methods are overestimating the true values of the test set (right part of Figure 4.33). One could think about a possible

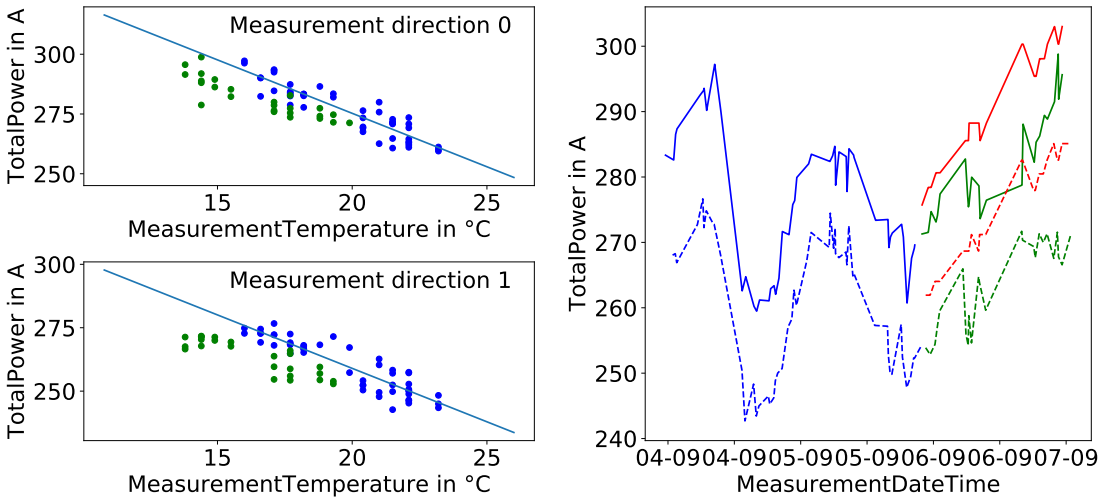

Figure 4.33: Training set (blue), test set (green), estimated regression lines on the left and the 25-step linear regression forecasts (red) on the right

solution taking enough data points before the test set of each measurement direction until the non-linearity becomes apparent and estimate the coherence between MeasurementTemperature and TotalPower based on an exponential model, more specifically, by

$$y_t = c_0 + c_1 \exp(-c_2 t). \tag{4.3}$$

We want to analyse this approach with enough data points to estimate the coefficients c_0, c_1, c_2 by non-linear least squares[4]. Based on the 11887 data points of the year before the last test set, we estimate for each measurement direction the coefficients of (4.3) and perform forecasts based on the estimated curves. With this approach, we can improve the 50-step forecast (25-step forecasts for each direction) among all forecasting methods for the last test set. We get the following output of the MSEs:

```
Mean squared error of the 1-step forecast:  4.6
Mean squared error of the 10-step forecast:  6.3
Mean squared error of the 50-step forecast:  25.0.
```

The estimated function of each measurement direction and the 50-step forecast can be seen in Figure 4.34.

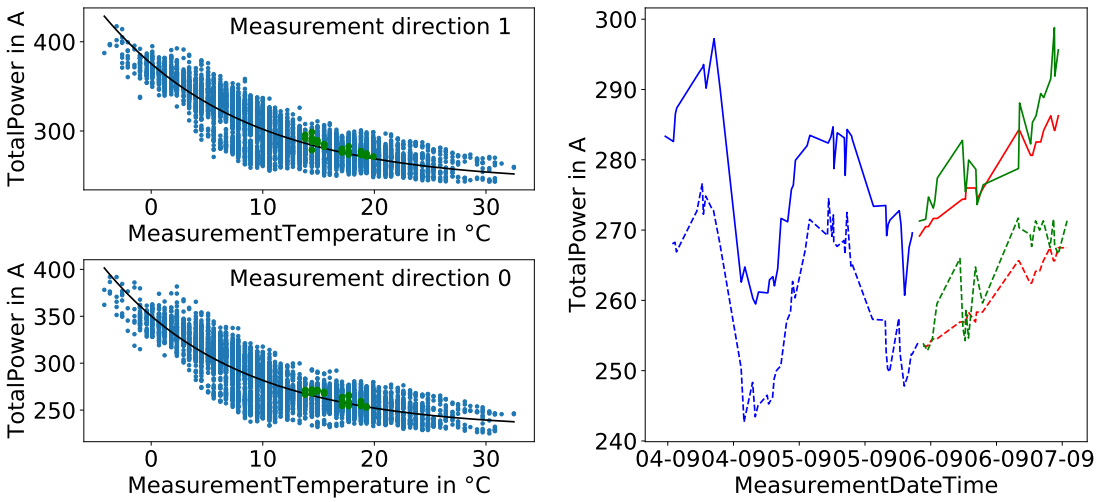

Figure 4.34: Training set (blue dots), test set (green dots) and estimated exponential curves (black) on the left. On the right the last part of the training set (blue), the 10th test set (green) and the 25-step forecasts (red)

[4]Using optimize.curve_ fit of SciPy

Although this seems beneficial, we cannot improve the best 1-step and 10-step forecast. A big problem is that we need a lot of data points and it may be that the test set does not lie on or near the exponential curve and behaves linearly, for example, if we take the 11217 data points of the year before the 5th test set and go on as before we get worse MSEs in comparison to the linear and local models:

```
Mean squared error of the 1-step forecast:  218.8
Mean squared error of the 10-step forecast:  306.7
Mean squared error of the 50-step forecast:  262.5
```

The estimated curves and the forecast can be seen in Figure 4.35. The global exponential method using the data points of the last year had the problems of overfitting. We conclude that one should rather use local models because the electrical consumption of the railway switch has a big variance over a year and can change dramatically (e.g. through maintenance action).

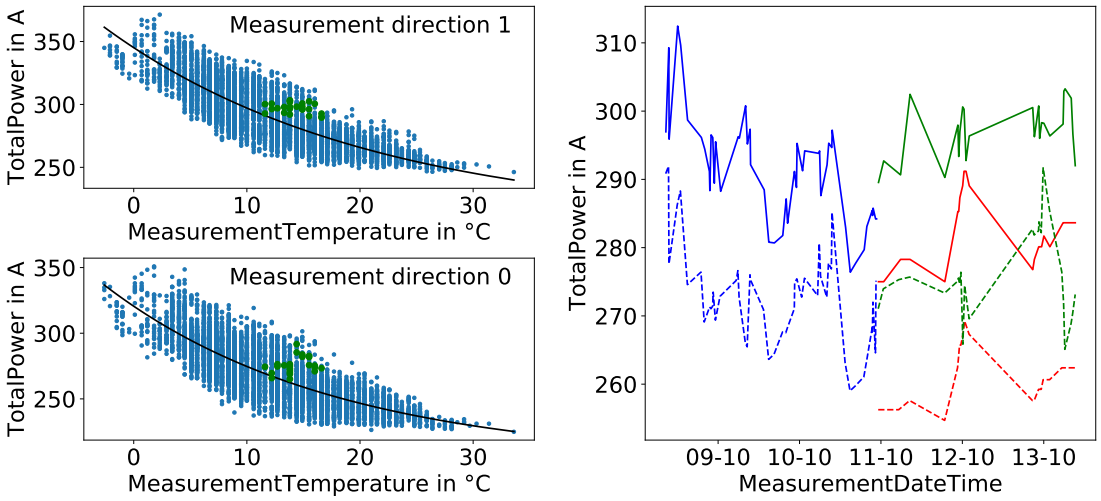

Figure 4.35: Training set (blue dots), test set (green dots) and estimated exponential curves (black) on the left. On the right the last part of the training set (blue), the 5th test set (green) and the 25-step forecasts (red)

4.3 Switch 3

The data of Switch 3 consists of 4331 observations recorded from 15-07-2019 to 12-09-2019 and 9 variables (MeasurementDateTime, MeasurementTemperature, MeasurementDirection, Multiplier, offset, MeasurementDataFormat, MeasurementData, DataStart, DataEnd). The variables have the same meaning as in the previous switches. As before we create a new variable "TotalPower" which represents the approximation of the total power by adding up all values of "MeasuredData". Additionally, we have data from local temperature and humidity measurements provided in 15-minute intervals, starting from 15-07-2019 06:15:01 to 06-09-2019 06:00:00 except for two measurements on the 25-08-2019 at time 06:00:00 and 06:00:03. Also, we have data of local temperature and humidity measurements provided in 5-minute intervals starting from 06-09-2019 11:05:00 to 12-09-2019 06:00:00. The data of local temperature and humidity measurements consists of 9 variables (Timestamp, TZ, T Bottom (degC), T LB HOT (degC), T LB COLD (degC), Temp outdoor (deg C), Humidity outdoor (%RH), Temp inside (deg C), Humidity inside (%RH)). The local measurements of interest have the following meanings:

- "Timestamp": Time of the measurement
- "T Bottom (degC)": Temperature directly below the point machine
- "T LB HOT (degC)": Temperature of the rail, near the switch heating (which is operated during cold weather)
- "T LB COLD (degC)": Temperature of the rail (not near the switch heating)
- "Temp outdoor (deg C)": Temperature near the switch at 1m above the ground
- "Humidity outdoor (%RH)": Humidity near the switch at 1m above the ground
- "Temp inside (deg C)": Temperature inside the point machine
- "Humidity inside (%RH)": Relative humidity inside the point machine

We merge the two data sets together. For each switch movement, we add the values of the local temperature and humidity measurements by taking the values which are closest in time and adding them to the 4331 observations of the data set of the switch movements. This means the local temperature and humidity measurements have a maximum inaccuracy of 7.5 minutes until the 06-09-2019 06:00:00

and a maximum inaccuracy of 2.5 minutes from the 06-09-2019 11:05:00 to the 12-09-2019 06:00:00 for each switch movement. In the gap on the 06-09 where we have no data for local temperature and humidity measurements provided, 21 recorded switch movements have taken place. It follows that for those records we have a maximum inaccuracy of about 2.5h. The last local measurements of local temperature and humidity were on the 12-09-2019 06:00:00. After this time we have 37 switch movements recorded until 15:00:30 on the same day which still uses the local temperature and humidity measurement of 06:00:00. There are 21 missing values of the variable "Humidity outdoor (%RH)" on the 26-08. We delete the corresponding values and end up with the final data set of 4310 observations. The time series of TotalPower can be divided into 4 parts (c.f. Figure 4.36 and Table 4.19). No failures or maintenance actions have been reported for this switch. This

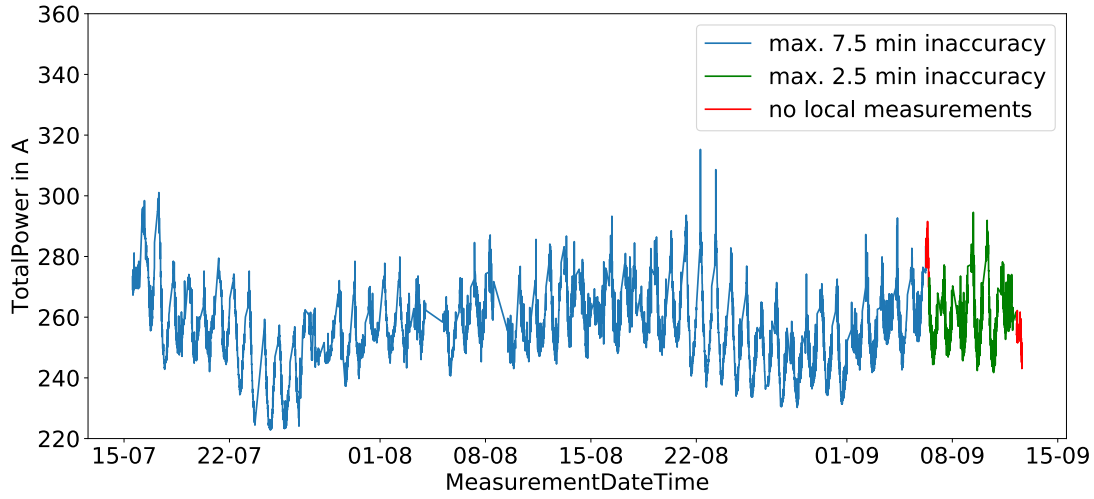

Figure 4.36: TotalPower plot (different parts of the local measurements colourised)

allows us to perform a rolling forecast. We want to use a usual (multiple) linear regression model[5] to find out which variables have a big influence on the total power consumed by the switch. First, we will use 100 data points as the training set (50 for each direction) and the following 50 data points as the test set (25-step forecasts for each direction). Then we will shift the training and test set 50 data points to the right and start all over again. A first relation between the 8 influencing variables of TotalPower and the target variable TotalPower can be seen in the sample autocorrelation matrix (Figure 4.37). In the last row, we can observe a negative correlation

[5]Using the ARMA model with exogenous inputs of Statsmodels with parameters (0,0)

Data set	Max. 7.5 min. inaccuracy	Gap with no local measurements	Max. 2.5 min. inaccuracy	End with no local measurements
Time frame	from 15-07-2019 13:30:16 to 06-09-2019 05:53:40	from 06-09-2019 06:01:13 to 06-09-2019 11:00:12	from 06-09-2019 11:17:03 to 12-09-2019 05:53:10	from 12-09-2019 06:00:29 to 12-09-2019 15:00:30
Number of data points	3811	21	441	37

Table 4.19: Different sections of Switch 3

between TotalPower and the temperature variables along with a positive correlation between TotalPower and the humidity variables. The two sample correlations between a humidity variable and TotalPower are comparable among each other just as the sample correlations between a temperature variable and TotalPower.

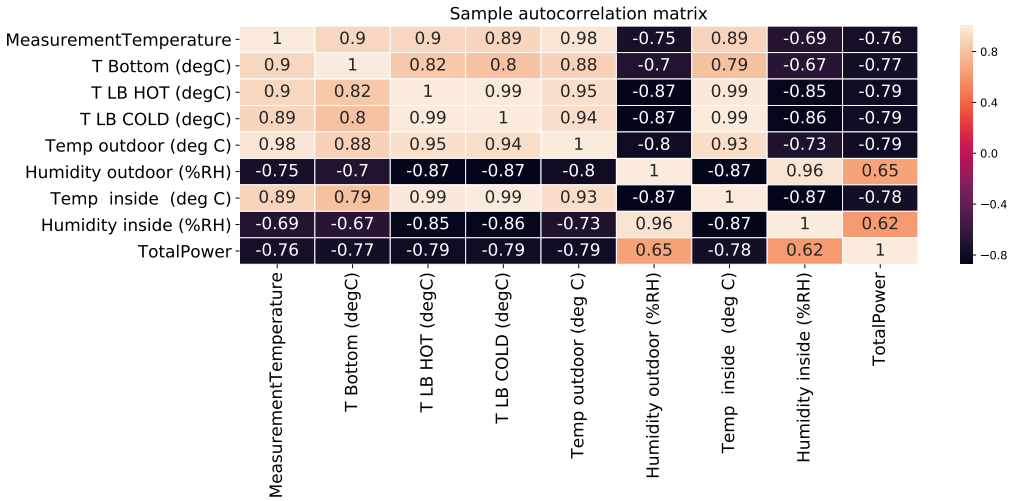

Figure 4.37: Autocorrelation matrix of the influencing variables and TotalPower

4.3.1 Forecasts based on different measured temperatures

Looking at the progression of "MeasurementTemperature" (Figure 4.38), we observe that outliers are still present. Also at the bottom plot of Figure 4.38, we can see that many data points at about 10 °C are strongly separated from the others.

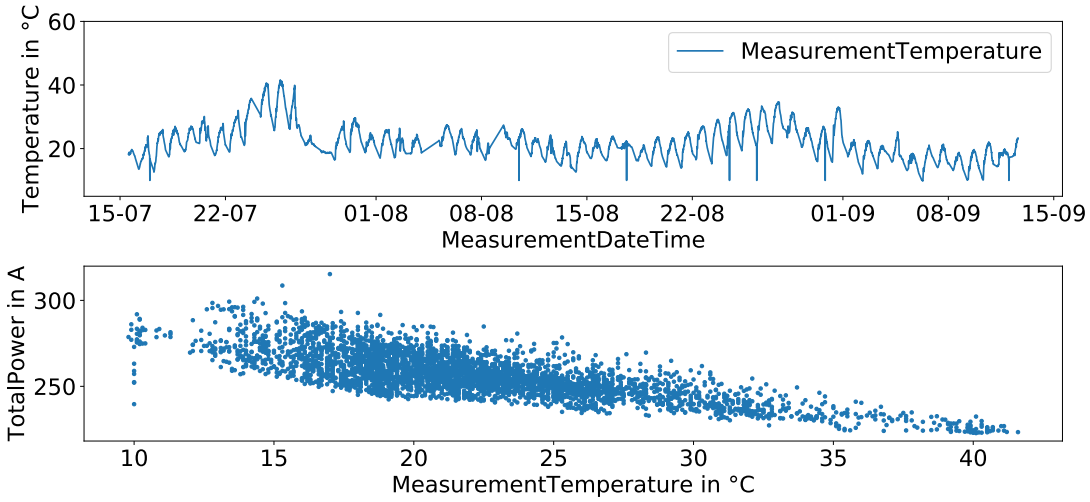

Figure 4.38: Progress of the variable MeasurementTemperature (top) and MeasurementTemperature to TotalPower (bottom)

We want to choose the forecasting horizon so that there are no outliers of the variable MeasuredTemperature present to compare the forecasting results with the results we obtain with other local measured temperatures. We pick a smaller time window to test the forecasts including different temperatures against each other. We want to begin forecasting on the 21-08-19. That means we choose the 100 data points before as the training set to perform the first 50-step forecast. After that, we shift the training and test set 50 data points to the right. We do this 5 times which brings us to the 2019-08-24 09:54:55. The time series can be seen at the top of Figure 4.39. At the bottom of Figure 4.39, the timely progress of the 6 measured temperatures of the training and forecasting time frame can be seen. We notice that the temperature near the switch at 1m above the ground (Temp outdoor (degC)) can be compared to the measured temperature at the relay house nearby. The same applies to the temperature directly below the point machine (T Bottom (degC)). However, it seems that "T Bottom (degC)" lags a little behind "MeasurementTemperature". By comparison, the other measured local temperatures "T LB HOT (degC)" and "T LB COLD (degC)" which measure the temperature of the rail and "Temp inside (degC)" which measures the temperature inside the point machine are mostly lower at night and much higher during the day and also in Figure 4.40 we notice that the temperatures measured inside the point machine and on the railway can be much higher compared to the others. It is clear that "T LB HOT (degC)"

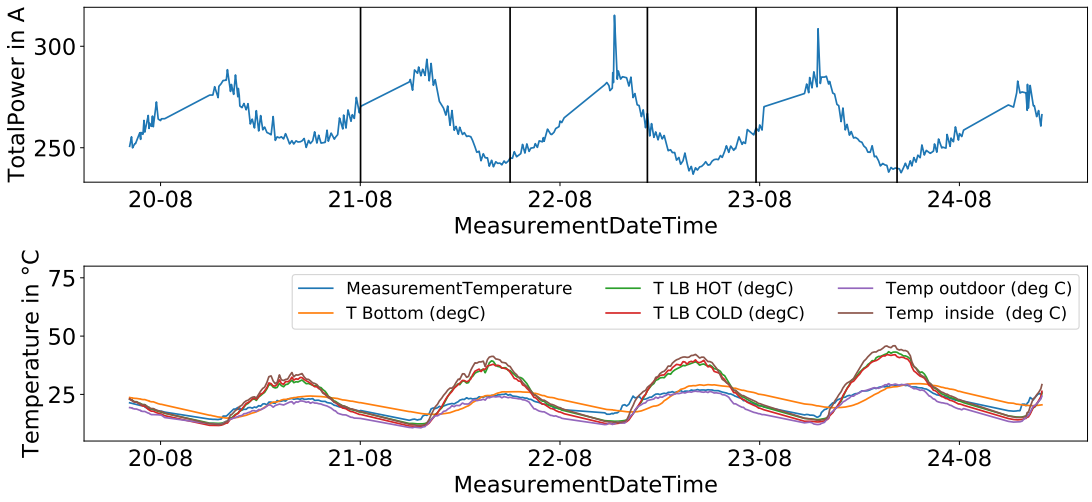

Figure 4.39: At the top the time series we want to analyse (the 5 forecasting windows separated by black vertical lines) and the progress of the different temperatures at the bottom

and "T LB COLD (degC)" behave almost identically because in summer there was probably no heating of the railway. After looking at the values of TotalPower in this time frame and separating it into two measurement directions we can see two peaks of TotalPower for the measurement direction 1 (top of Figure 4.41) recorded on the 22nd and 23rd August.

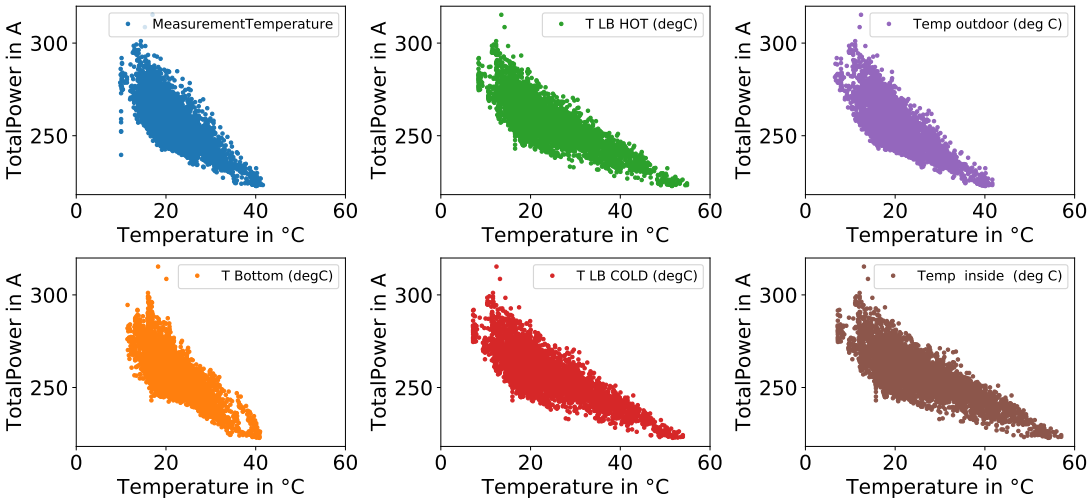

Figure 4.40: Temperature to TotalPower plot of each temperature variable

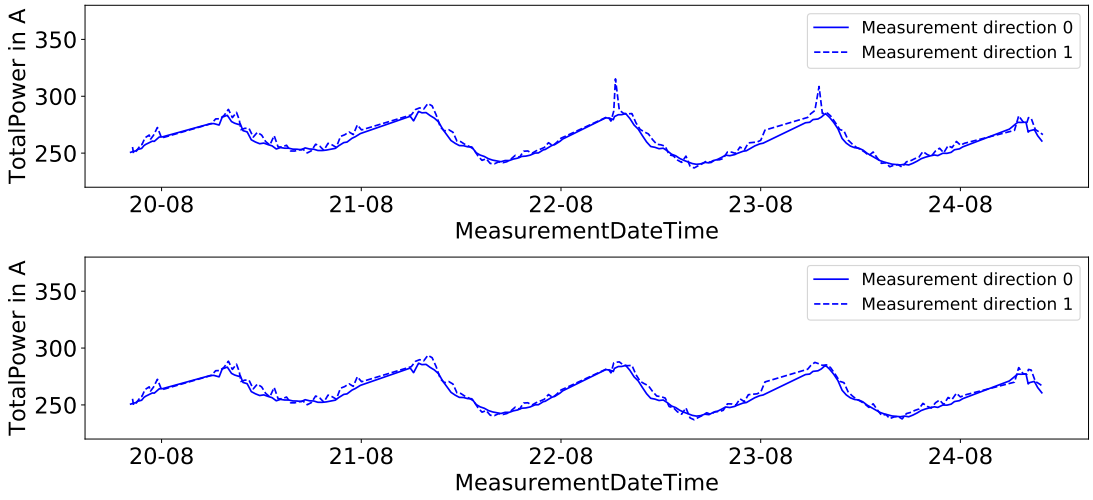

Figure 4.41: The time series divided into different directions (top) and with replaced outliers (bottom)

We identify those peaks as outliers because no extreme temperatures were measured in those two timestamps. These outliers would have an influence for computing the MSE for some forecasts and even for the coefficient estimation of the last models of the rolling forecast. We eliminate this by replacing the two values at the respective location (lok) by taking the mean of the two values of the measurement direction nearby, i.e. $x^1_{\text{lok}} = \frac{1}{2}(x^1_{\text{lok}-1} + x^1_{\text{lok}+1})$. The corrected set, which we will use for training and forecasting can be seen at the bottom of Figure 4.41. However, we have not seen the progress of humidity at this time yet, which might be extreme. If this were the case and/or the predictions for the measurement direction 1 would shot up in one of those two timestamps of the removed outliers we need to rethink the approach as treating the two values as outliers. Unlike the two other switches, there is no big gap between the two measurement directions detectable. Looking at Table 4.20 we see that the MSEs of the different linear regression forecasts are similar. The MSEs of the whole forecasting window is listed in the column "forecasts combined". Using the variable "T Bottom (degC)" as exogenous variable made worse forecasts in most cases. The best rolling forecast in terms of the MSE produced in this situation the variable "T LB HOT (degC)" illustrated in Figure 4.42.

Variable \ Test set	1	2	3	4	5	forecasts combined
MeasurementTemperature	60.1	147.9	**23.2**	52.7	56.7	68.1
T Bottom (degC)	100.9	62.1	89.7	104.3	56.2	82.6
T LB HOT (degC)	53.7	**56.6**	27.1	55.6	41.6	**46.9**
T LB COLD (degC)	54.8	71.7	39.1	**48.5**	50.1	52.9
Temp outdoor (deg C)	**35.2**	79.3	40.8	71.8	**38.8**	53.2
Temp inside (deg C)	72.2	74.4	38.9	66.5	52.8	61

Table 4.20: MSEs of the 50-step forecasts including a temperature variable of the 5 forecasting windows and the respective MSE of the rolling forecast

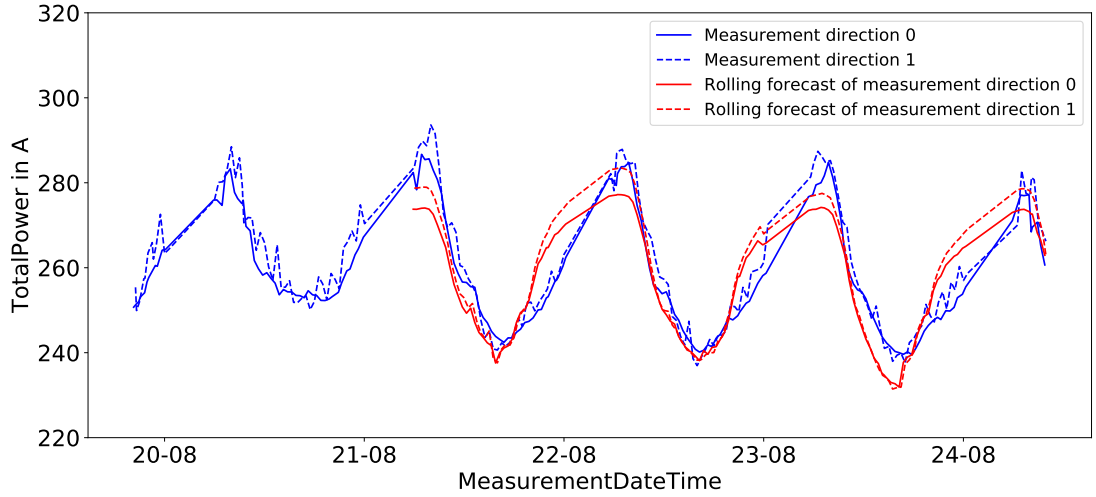

Figure 4.42: Rolling 5x 50-step linear regression forecast containing the variable "T LB HOT (degC)" as regressor

Now we want to test if we can optimise the forecasts through the addition of more regressors. We choose k as the number of regressors and add a constant $c \in \mathbb{R}$. The forecasting model now has the form

$$X_t = c + \eta_1 U_{1t} + \ldots + \eta_k U_{kt} + \epsilon, \tag{4.4}$$

where $\eta_1, \ldots, \eta_k$ are scalars of the different external time series $U_{1t}, \ldots, U_{kt}$ (in this case the different temperatures) and ϵ is the error term. In Table 4.21 we can see the MSEs of some combinations of the temperatures. Combining two variables that produced good forecasts ("T LB HOT (degC)" and "Temp outdoor (deg C)") produced similar results as the best combined forecast containing only one single

variable (T LB HOT (degC)). The MSE of the rolling forecast improves a little if we use all 6 temperature variables as regressors. Removing the poorest performed variable (T Bottom (degC)) from the set of all temperature variables did not result in a better forecasting result than all temperature variables combined. There are $\sum_{i=1}^{6} \binom{6}{i} = 63$ possible regressor combinations. The combination that gave us the best rolling forecast is "'T Bottom (degC)', 'T LB HOT (degC)'" with a MSE of 27.6 (whole forecasting window) among all possible combinations. The worst rolling forecast for the combinations with the temperature variables was the one with only the single variable "T LB HOT (degC)".

Variables \ Test set	1	2	3	4	5	forecasts combined
T LB HOT (degC), Temp outdoor (degC)	42.8	63	27.6	55.4	42.5	46.2
T Bottom (degC), T LB HOT (degC)	**33.7**	**20.4**	38.5	**29**	**16.5**	**27.6**
All measured temperatures except T Bottom (degC)	29.8	54.9	**20.2**	109.1	46.7	52.1
All 6 measured temperatures	59	25.7	64.2	42.5	19.4	42.2

Table 4.21: MSEs of the 50-step forecasts containing multiple temperature variables as regressors of the 5 forecasting windows and the respective MSE of the rolling forecast

4.3.2 Forecasts that include humidity as an exogenous variable

At the top of Figure 4.43 we can see the chronological sequence of the two local measurements of humidity. In most weekly time frames, we observe a daily season. In those cases, both variables achieve a maximum of relative humidity in the morning and a minimum in the afternoon. It is mostly recognisable that the relative humidity measured outside the point machine reaches higher maxima than the humidity measured inside the point machine whereas the humidity measured inside the point machine reaches lower minima than the humidity measured outside the point machine. At the bottom of Figure 4.43, we can see that the variance of TotalPower rises with the increase in relative humidity measured outside. In the case

of the humidity measured inside the point machine, it seems that the variance is also increasing at first but decreases again towards the end and also the values of TotalPower are decreasing with higher relative humidity at about 80%. No extreme

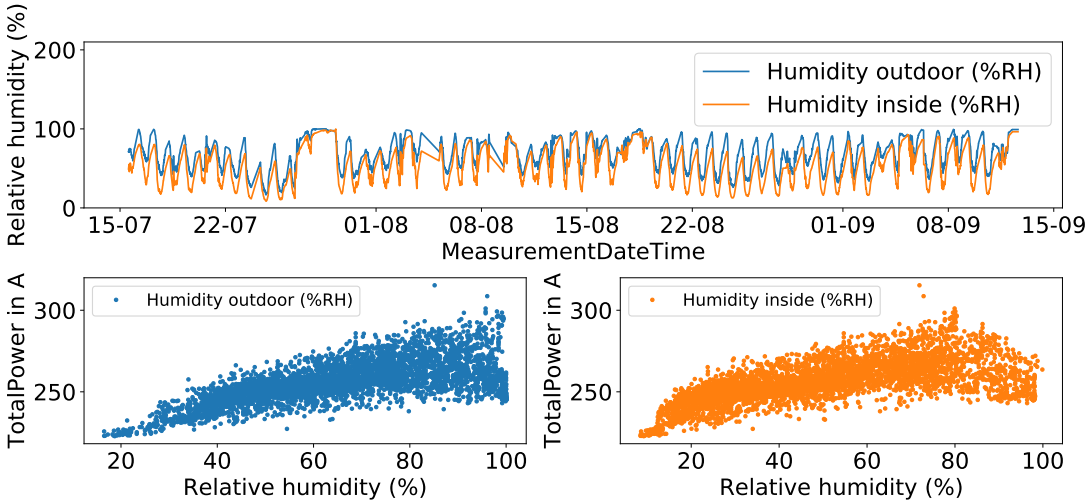

Figure 4.43: The progress of relative humidity (top) and plots of the humidity variables to TotalPower (bottom)

value for relative humidity is recognizable at the bottom of Figure 4.44 at the time we replaced the outliers of TotalPower (red dots), confirming our suspicion treating the original values as outliers.

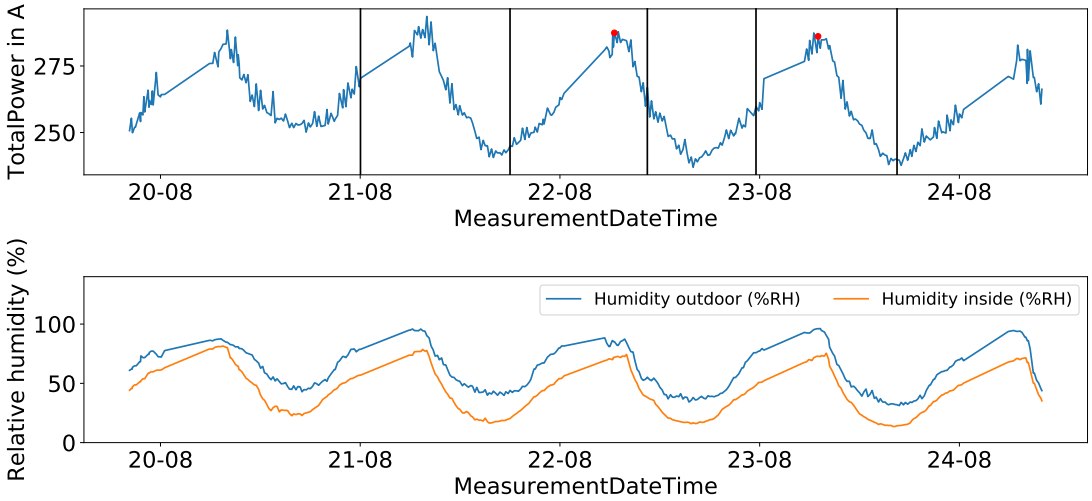

Figure 4.44: TotalPower with replaced outliers (top) and the progress of relative humidity (bottom)

Table 4.22 shows the MSEs of the different combinations of humidity as exogenous variables. Taking "Humidity inside (%RH)" as an exogenous variable produced mostly better forecasts than using the variable that measures the humidity outside the point machine and an even better forecast than the best rolling forecast that includes a single temperature variable as a linear regressor. A combination of the two humidity measurements turns out a similar result.

Variable(s) \ Test set	1	2	3	4	5	forecasts combined
Humidity outdoor (%RH)	**54.1**	80.7	32.2	29.6	53.6	50
Humidity inside (%RH)	67.4	**44.1**	**21.4**	**19.4**	36.4	**37.7**
Humidity outdoor (%RH), Humidity inside (%RH)	60.8	47.3	30.4	24.4	**35.1**	39.6

Table 4.22: MSEs of the forecasts containing a single and combined humidity variables as regressors and the respective MSE of the rolling forecast

Modeling TotalPower with a combination of two variables, one of temperature and one of humidity that gave us the best forecasts individually (Figure 4.45) did not result in a better forecast than only using "Humidity outdoor(%RH)" (Table 4.23).

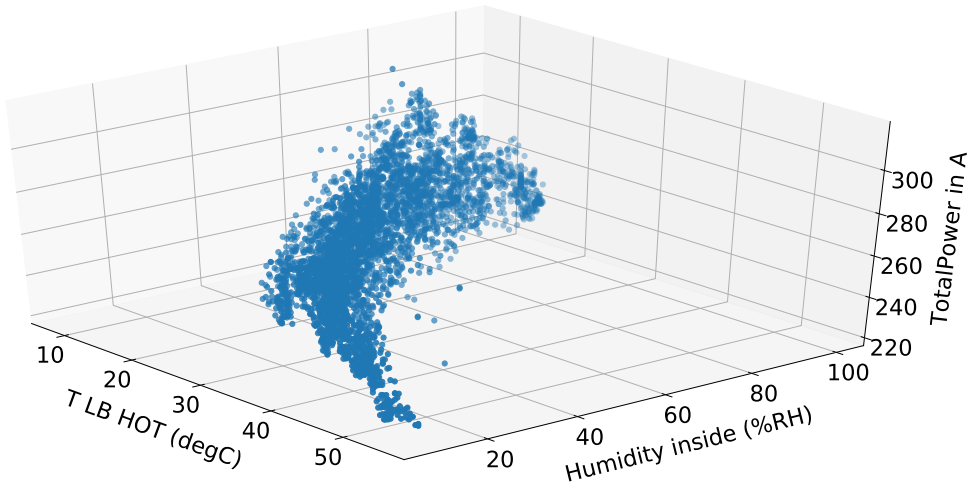

Figure 4.45: A 3d plot consisting of temperature and humidity to TotalPower

After a grid search of all 255 possible combinations, it turned out that using only "T Bottom (degC)" still produced the worst rolling forecast. The best combination was

"'T Bottom (degC)', 'T LB COLD (degC)', 'Temp outdoor (deg C)', 'Humidity outdoor (%RH)', 'Humidity inside (%RH)'" with a MSE of 15.1 (Table 4.23). The worst rolling forecast, containing only "T LB COLD (degC)" can be seen at the top of Figure 4.46. The lagged property of this variable has the same effect on the rolling forecast and therefore it should not be used in a linear regression forecast which contains only a single exogenous variable. However, "T LB COLD (degC)" appears in the combination of the best regressor variables of temperature as well as in the best combination of all variables and is of value when combined. At the bottom of Figure 4.46, we can see the forecast of the best combination of variables. The lagged property of "T LB COLD (degC)" no longer negatively affects the forecast.

Variables \ Test set	1	2	3	4	5	forecasts combined
T LB HOT (degC), Humidity inside (%RH)	93.0	48	21.1	15.6	37.6	43.1
T Bottom (degC), T LB COLD (degC), Temp outdoor (deg C), Humidity outdoor (%RH), Humidity inside (%RH)	**25.6**	**10.1**	**10.9**	**11.7**	**17.1**	**15.1**

Table 4.23: MSEs of the forecasts containing humidity and temperature variables and the combination of the best rolling forecast

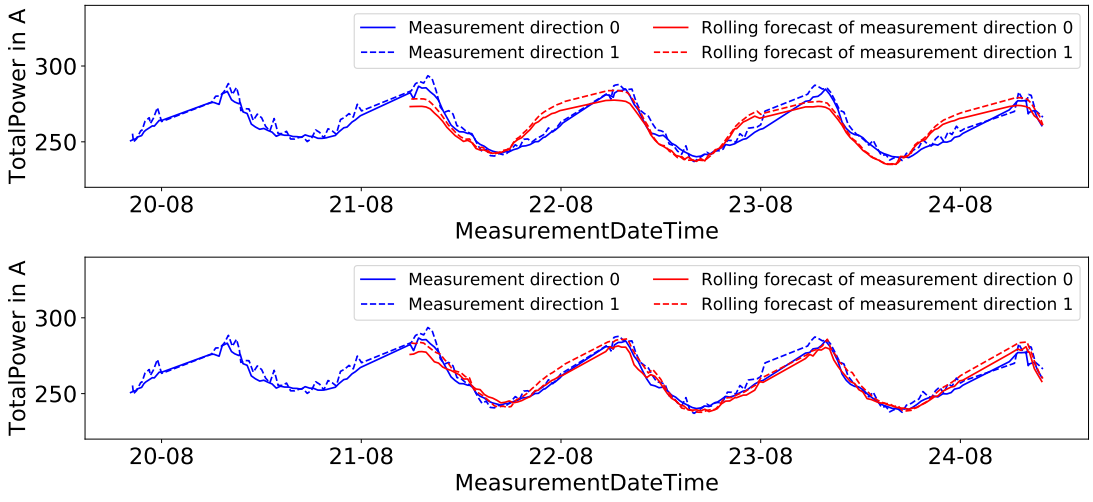

Figure 4.46: The worst rolling forecast (top) and the best rolling forecast (bottom)

5 Summary and outlook

In the evaluations of the three switches, it became apparent that the measurement direction plays an important role when forecasting the time series of the (discrete) total power. One challenge that occurs with all discussed models is how to handle gaps between measurements. One of the main problems that arise when forecasting the (discrete) total power with ARMA and extended ARMA models is that they are defined on whole numbers, but the time interval between two switch movements is not equidistantly distributed. Nevertheless, ARIMA models gave very good 1-step predictions and could be used to model the total power at a current time point of a frequently used railway switch. The ARIMA models were mostly superior to the other models that included exogenous variables for 1-step forecasts. The ARIMA model can detect the alternating dependence on the measurement direction, even when choosing the parameter $d = 0$. However, the procedure to choose the parameter d with the augmented Dickey-Fuller test or based on the smallest AIC value of the grid search mostly resulted in the parameter $d = 1$. That is reasonable because of the dependence of the total power on the measurement direction and therefore to choose an ARIMA model is advisable compared to choosing only an ARMA model. In the chapter of Switch 1, the 1-step predictions of the VARMA model were better than the 1-step ARIMA forecasts in all five test sets whereas in the evaluation of Switch 2 this was not the case. That is probably caused because the gap (i.e. the difference of the (discrete) total power between the two measurement directions) was bigger in the first switch than in the analysed switch movements of the second switch. Therefore, this gap between the measurement directions must be considered when choosing whether to use an ARIMA or VARMA model. These two models can be used for forecasting a small-time horizon to model the current state of the switch. If the actual measured value extends beyond a given threshold it can be concluded that a failure has occurred. This may work with failures that occur suddenly and which are reflected in an increase in total power. It emerged that

the (discrete) total power consumed by the switch motor is strongly dependent on external influences such as temperature. If for example the training set for parameter and coefficient determination of an ARIMA model is taken by day and the time of the next switch movement that one would like to forecast is in the night with a much lower temperature the ARIMA/VARMA model may not provide a good value. The problem that arises with models that do not include exogenous inputs is if the training/test set is chosen at different time points where the temperature and other factors that influence the total power consumption vary a lot and are not repeated the same. Especially for vector-valued models, where the prediction of one direction is also dependent on values of the other direction, the time between the switch movements is crucial because of the variation of temperature and other exogenous factors. For multi-step forecasts, it is not advisable to use an ARIMA model. It can get completely out of hand and produce forecasts that are getting further and further away from the true values (c.f. Figure 4.27). The VARMA model prevents this by converging to a constant for each direction but it is also not able to predict the changes of total power over time, influenced by external factors. It is recognised that for a multi-step forecast, the functional relation between the (discrete) total power and external factors needs to be considered.
An ARIMA/VARMA model can be used to expose a sudden malfunction of a frequently used switch with the characteristics discussed above but they are not capable of detecting degrading effects. Reconsidering the forecasts including exogeneous variables of the second shifted data set of the first switch, all models that included temperature as exogenous input did not forecast the ups and downs of TotalPower (such as the linear regression forecast at top of Figure 5.1) and we classified the behaviour as abnormal. Modeling the time frame with a rolling 1-step AR(1) process that gets updated after each step models the misfunctioning of the switch what is not reasonable (c.f. bottom of Figure 5.1). For recognition of degrading effects, multi-step forecasts are needed. Also, the model needs to be updated from time to time with local information of the past. In this thesis 100 data points were enough to model the different characteristics such as temperature that occur in one day and locally based forecasts were superior to the global one that had the problem of overfitting. It turned out that in small time frames a linear regression forecast for each direction is able to describe the total power consumption of the analysed railway switch. To model the error as an ARMA process was mostly of advantage for 1-step forecasts. As a regressor temperature but also humidity and

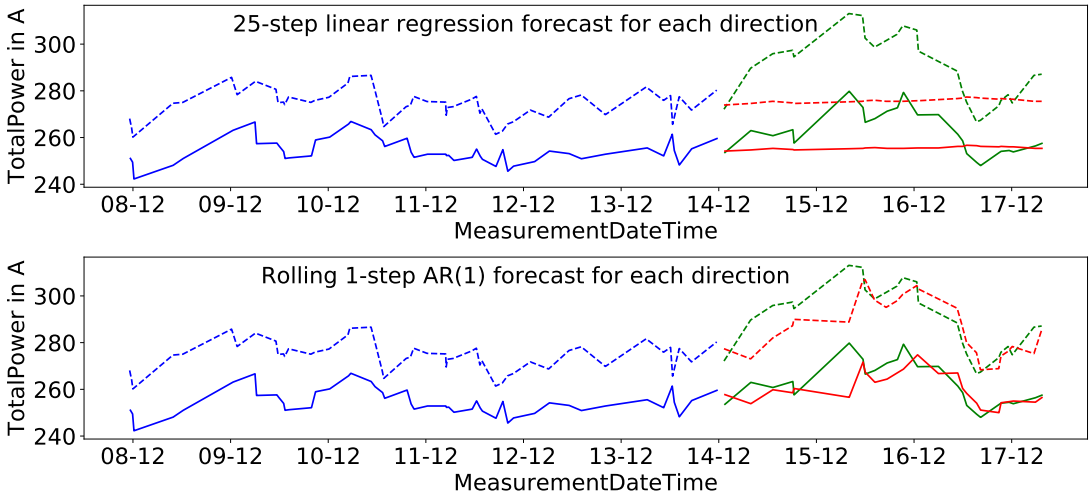

Figure 5.1: Training set (blue), test set (green) and forecasts (red)

combinations of local measured temperature and humidity regressors can be used. Daily use could be to train the model for each direction with 50 data points and forecast 25 data points for each direction. After forecasting, the training and test set can be shifted 25 data points in each direction to produce a rolling forecast to obtain an image of the total power consumption as done in chapter 3 (c.f. Figure 4.46). A method to detect a failure or degrading effects could provide a confidence interval for each forecast (e.g. for the shifted test set of the first switch, Figure 5.2).

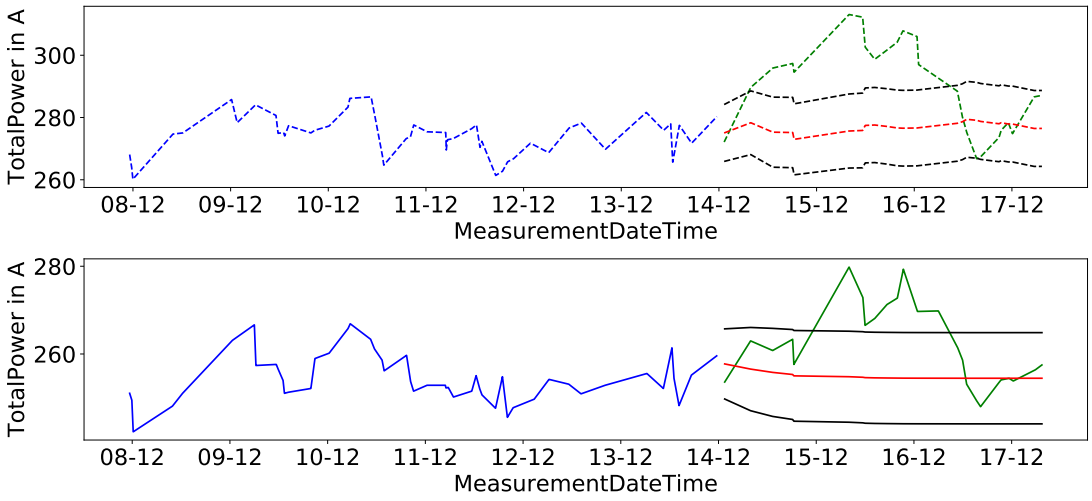

Figure 5.2: Training set (blue), test set (green) and confidence intervals (black) of the 25-step linear regression forecasts (red) for each direction

An alert can be triggered when the true value exceeds the confidence interval (multiple) times. Because of the high dependence on external factors, there must be exact measurements of temperature, humidity and so on. They must be measured precisely during each switch movement to create a suitable model for an accurate image of the condition of the switch motor on the basis of total power. The exact relation between temperature, humidity and consumed total power still needs to be studied and is not yet clear. Also other approaches, for example in Machine Learning need to be considered, to see if they are even more suitable for describing the total power consumption of a switch using several variables as inputs. Errors and unusual behaviour of the switch that can occur need to be classified. It has to be found out how which error or malfunction is indicated as well as when and how they become noticeable in the total power consumption of a railway switch.

A Appendix

The forecasts are based on a training set of 50, 150 and 200 data points before the respective test set. The parameters of the models of the first two switches were calculated based on the smallest AIC. The forecasts of Switch 3 were made with the variables of the chapter as well as the best and worst combinations for temperature variables and the best and worst combinations of all influencing variables (whole rolling forecasting window).

A.1 Forecasts with other sample sizes of Switch 1

A.1.1 Forecasts based on a training set of 50 data points

Method \ Test set	1	2 shifted	3	4	5
ARIMA	(4,1,4)	(4,1,1)	(1,1,3)	(3,0,1)	(2,1,2)
VARMA	(0,2)	(0,2)	(0,4)	(2,0)	(0,1)
VARMAX	(1,0)	(1,0)	(1,0)	(1,1)	(1,0)
Liner regression with ARMA errors	0-(2,0) 1-(2,1)	0-(0,4) 1-(4,2)	0-(0,1) 1-(0,0)	0-(2,0) 1-(0,3)	0-(0,3) 1-(0,3)

Table A.1.1: Parameters of the estimated models based on 50 data points of the first switch

Method \ Test set	1	2 shifted	3	4	5
1-step forecasts					
ARIMA	32.6	23.4	7.3	0.9	2.9
VARMA	**0.1**	4.1	13.7	0.2	15.4
VARMAX	0.2	43.4	2.6	1	0.2
Linear regression with ARMA erros	1.4	9.2	**1.1**	**0**	**0.1**
Linear regression	3.5	**2.6**	9.6	0.1	8.4
10-step forecasts					
ARIMA	58.8	**109.1**	32.3	13.4	51.5
VARMA	18.3	235.9	27.6	**4.8**	10.2
VARMAX	**16.6**	164.5	**20.1**	14.8	**9.8**
Linear regression with ARMA erros	26.4	126.4	20.8	11.4	23.6
Linear regression	25.6	169.6	21.6	7.3	12
50-step forecasts					
ARIMA	111.2	198.5	**94.5**	53.9	436.6
VARMA	33.3	336.2	100	**44**	172.1
VARMAX	**23.9**	265.7	123.8	58.4	**73.6**
Linear regression with ARMA erros	28.6	**219.5**	124.7	52.3	105.1
Linear regression	28.8	258	124.8	45.8	103

Table A.1.2: Forecasts based on a training set of 50 data points of the first switch

A.1.2 Forecasts based on a training set of 150 data points

Method \ Test set	1	2 shifted	3	4	5
ARIMA	(4,0,1)	(3,1,3)	(1,1,4)	(2,1,1)	(2,0,1)
VARMA	(2,0)	(1,0)	(1,0)	(1,0)	(0,4)
VARMAX	(1,3)	(1,0)	(1,0)	(1,0)	(1,0)
Liner regression with ARMA errors	0-(4,2) 1-(3,2)	0-(1,0) 1-(0,2)	0-(4,4) 1-(2,2)	0-(0,2) 1-(4,0)	0-(0,3) 1-(0,2)

Table A.1.3: Parameters of the estimated models based on 150 data points of the first switch

Method \ Test set	1	2 shifted	3	4	5
1-step forecasts					
ARIMA	7.9	30.4	10.7	3.6	**0**
VARMA	6.9	23.9	**6.3**	0.3	10.5
VARMAX	11.2	25.4	10.1	12.5	**0**
Linear regression with ARMA erros	**3.2**	5.8	19.9	**0.2**	4.9
Linear regression	11.1	**0.5**	27.4	7.3	2.7
10-step forecasts					
ARIMA	17.2	**158.7**	31	29.2	8
VARMA	21	185.5	27.3	11.8	10.2
VARMAX	**20**	175.6	**20.5**	11.5	5.1
Linear regression with ARMA erros	28.3	189.4	21.4	**10**	**2.3**
Linear regression	24.2	201.8	24.3	11.4	4.6
50-step forecasts					
ARIMA	31.9	**271**	**88.8**	156.2	130.4
VARMA	32.8	302.4	105	**42.6**	134.4
VARMAX	51.9	283.9	158.1	46.3	**60.4**
Linear regression with ARMA erros	29.2	291.6	129.1	52	72.1
Linear regression	**25.7**	296.3	143.3	47.2	71

Table A.1.4: Forecasts based on a training set of 150 data points of the first switch

A.1.3 Forecasts based on a training set of 200 data points

Method \ Test set	1	2 shifted	3	4	5
ARIMA	(3,1,4)	(4,1,2)	(2,0,1)	(4,0,2)	(2,0,1)
VARMA	(3,0)	(2,0)	(1,0)	(2,0)	(2,0)
VARMAX	(2,0)	(1,0)	(1,0)	(2,0)	(1,0)
Liner regression with ARMA errors	0-(1,1) 1-(4,0)	0-(1,0) 1-(1,1)	0-(1,0) 1-(1,1)	0-(0,2) 1-(3,2)	0-(0,2) 1-(2,0)

Table A.1.5: Parameters of the estimated models based on 200 data points of the first switch

Method \ Test set	1	2 shifted	3	4	5
1-step forecasts					
ARIMA	21.4	38.7	9.5	2.9	**0**
VARMA	6.8	15.9	**5.8**	**0**	6.6
VARMAX	**2.1**	27.7	8.9	3.7	0.1
Linear regression with ARMA erros	3.6	12.8	6	**0**	1.2
Linear regression	43.3	**4.1**	27.7	4.3	0.6
10-step forecasts					
ARIMA	66.5	**138.8**	29.5	36.8	8.2
VARMA	25.1	188.4	29.7	**10.4**	10
VARMAX	**14**	195.8	21.7	12.8	3.2
Linear regression with ARMA erros	41.1	191.8	**20.7**	24	3.3
Linear regression	28.6	250.6	21.9	12.2	**2.7**
50-step forecasts					
ARIMA	106.5	**232.6**	100.1	147.8	129.1
VARMA	42.4	340.2	**98.6**	45.4	132.7
VARMAX	50.6	330.8	137.3	**43.7**	**49.7**
Linear regression with ARMA erros	36.8	339	144.8	56.8	69.5
Linear regression	**21.8**	339.7	136	47.3	63.9

Table A.1.6: Forecasts based on a training set of 200 data points of the first switch

A.2 Forecasts with other sample sizes of Switch 2

A.2.1 Forecasts based on a training set of 50 data points

Method \ Test set	1	2	3	4	5
ARIMA	(1,1,1)	(3,0,2)	(3,1,2)	(1,1,1)	(1,1,1)
VARMA	(1,0)	(0,4)	(0,1)	(1,0)	(0,2)
VARMAX	(1,0)	(1,0)	(1,0)	(1,0)	(1,0)
Liner regression with ARMA errors	0-(2,2) 1-(2,3)	0-(0,1) 1-(1,0)	0-(0,0) 0-(0,0)	0-(1,0) 1-(1,0)	0-(3,1) 1-(0,3)

Table A.2.1: Parameters of the estimated models based on 50 data points of the first 5 data sets of the second switch

Method \ Test set	1	2	3	4	5
1-step forecasts					
ARIMA	2.5	22.1	1	0.2	15.4
VARMA	0.8	18.7	45.4	8.6	10.8
VARMAX	**0**	**15.2**	**0.2**	**0.1**	86.9
Linear regression with ARMA errors	0.6	156.3	12.3	2.9	16.2
Linear regression	5.7	126.8	12.3	15.2	**7.6**
10-step forecasts					
ARIMA	47.3	234.4	127.9	58.8	76.1
VARMA	**14.3**	234.6	46.4	79.7	**40.8**
VARMAX	51.1	76.7	**18.3**	27	66.1
Linear regression with ARMA errors	39	50.5	21.5	**10.2**	86.8
Linear regression	32	**30.2**	21.5	26.1	59.3
50-step forecasts					
ARIMA	168.4	132.4	442.2	89.8	152
VARMA	74.4	134.6	**33.3**	77.5	91.4
VARMAX	36.3	60.7	42.5	42.8	75.5
Liner regression with ARMA errors	43.7	63.8	38.4	29.3	75.7
Linear regression	**35.8**	**38.5**	38.4	**23**	**71.9**

Table A.2.2: Forecasts based on a training set of 50 data points of the first 5 data sets of the second switch

Method \ Test set	6	7	8	9	10
ARIMA	(1,1,1)	(3,2,3)	(2,1,0)	(3,2,4)	(1,1,1)
VARMA	(1,0)	(1,0)	(1,0)	(2,0)	(1,0)
VARMAX	(1,0)	(1,0)	(1,0)	(1,1)	(1,0)
Liner regression with ARMA errors	0-(1,0) 1-(1,0)	0-(1,0) 1-(0,0)	0-(4,0) 0-(1,0)	0-(0,1) 1-(0,1)	0-(1,0) 1-(2,0)

Table A.2.3: Parameters of the estimated models based on 50 data points of the last 5 data sets of the second switch

Method \ Test set	6	7	8	9	10
1-step forecasts					
ARIMA	1.5	**5.6**	0.3	26.8	1.2
VARMA	2.4	70.7	0.4	25.7	**0**
VARMAX	5.6	72.8	**0.1**	**20.1**	**0**
Linear regression with ARMA errors	**0.7**	73.2	6.8	57	2.9
Linear regression	3.9	101.8	20	41.1	19.9
10-step forecasts					
ARIMA	118.7	80.6	60	32.4	16.9
VARMA	77.8	71.5	91.2	103.5	**2.8**
VARMAX	78	20.5	36.4	**28.3**	12.9
Linear regression with ARMA errors	90.9	**19.3**	**18.5**	48.4	9.1
Linear regression	**64**	21.8	31.7	43.1	43.6
50-step forecasts					
ARIMA	721.7	576.4	207.1	540.2	370.3
VARMA	107.6	125.7	103.9	97.7	79.5
VARMAX	**81.3**	46.7	**26**	**32.4**	59.7
Liner regression with ARMA errors	88.2	**43.9**	67	45.6	**17.3**
Linear regression	86.5	44.8	31.5	42.4	75.3

Table A.2.4: Forecasts based on a training set of 50 data points of the last 5 data sets of the second switch

A.2.2 Forecasts based on a training set of 150 data points

Method \ Test set	1	2	3	4	5
ARIMA	(3,0,3)	(3,0,1)	(3,1,1)	(1,1,2)	(2,1,3)
VARMA	(1,0)	(1,0)	(1,0)	(1,0)	(1,0)
VARMAX	(3,0)	(1,0)	(1,0)	(1,0)	(1,0)
Liner regression with ARMA errors	0-(4,2) 1-(0,2)	0-(3,0) 1-(1,0)	0-(3,2) 0-(1,0)	0-(2,0) 1-(3,2)	0-(0,2) 1-(3,2)

Table A.2.5: Parameters of the estimated models based on 150 data points of the first 5 data sets of the second switch

Method \ Test set	1	2	3	4	5
1-step forecasts					
ARIMA	3.5	6.5	**3.7**	1.1	**0.3**
VARMA	9.2	**3.9**	31.3	9.4	3.5
VARMAX	1.9	20	21	**0.5**	13.2
Linear regression with ARMA errors	**0**	150.7	76.4	0.6	0.4
Linear regression	1.1	157.3	89	7.2	9
10-step forecasts					
ARIMA	66.5	344.5	72.5	63.3	45.8
VARMA	**19.3**	349	**36.3**	57.4	**14.1**
VARMAX	37.2	**28.7**	36.6	**9.5**	48
Linear regression with ARMA errors	57.7	110.5	77.3	15.9	34.2
Linear regression	46	152.3	110.9	13.2	44.1
50-step forecasts					
ARIMA	136.5	192.1	196.7	81	120.9
VARMA	116.1	193.6	55.4	76.5	**40.9**
VARMAX	**31.7**	103.3	**36.6**	67.1	63.7
Liner regression with ARMA errors	40.9	**40.4**	47.4	**44.8**	58.9
Linear regression	36.6	52.3	52.1	56.3	60.3

Table A.2.6: Forecasts based on a training set of 150 data points of the first 5 data sets of the second switch

Method \ Test set	6	7	8	9	10
ARIMA	(3,1,3)	(4,1,4)	(3,0,3)	(1,1,3)	(2,1,2)
VARMA	(1,0)	(1,0)	(1,0)	(4,0)	(1,0)
VARMAX	(1,0)	(1,3)	(1,0)	(1,0)	(1,0)
Liner regression with ARMA errors	0-(3,2) 1-(2,2)	0-(1,1) 1-(1,0)	0-(4,2) 0-(1,0)	0-(1,1) 1-(4,4)	0-(2,2) 1-(1,1)

Table A.2.7: Parameters of the estimated models based on 150 data points of the last 5 data sets of the second switch

Method \ Test set	6	7	8	9	10
1-step forecasts					
ARIMA	6.7	**19.6**	13.8	**6.2**	1.7
VARMA	3.9	105.4	1.7	13.9	0.2
VARMAX	**2.3**	120.3	13.4	16.4	**0**
Linear regression with ARMA errors	12.4	85.7	**0**	82.4	0.4
Linear regression	29.9	210.9	4.3	42.7	13.1
10-step forecasts					
ARIMA	70.5	81.6	53.6	43.4	18.2
VARMA	**69.5**	42.7	30.7	85.1	**13.1**
VARMAX	146.4	21	20.5	**38.2**	21.8
Linear regression with ARMA errors	173.1	**13.5**	**17.3**	51.6	20.7
Linear regression	123.3	30.6	22.3	43.4	72.7
50-step forecasts					
ARIMA	115.2	170.7	129	76.8	375.1
VARMA	**64.3**	105.9	119.1	120.3	42.8
VARMAX	169.6	**37**	85.3	**32.6**	53
Liner regression with ARMA errors	150.5	44.4	49.5	41.4	**34.4**
Linear regression	125	43.9	**38.8**	41.8	75.7

Table A.2.8: Forecasts based on a training set of 150 data points of the last 5 data sets of the second switch

A.2.3 Forecasts based on a training set of 200 data points

Method \ Test set	1	2	3	4	5
ARIMA	(3,0,3)	(2,1,4)	(3,1,1)	(2,1,3)	(4,1,1)
VARMA	(1,0)	(1,0)	(2,0)	(1,0)	(2,0)
VARMAX	(3,0)	(1,0)	(1,0)	(1,0)	(1,0)
Liner regression with ARMA errors	0-(0,2) 1-(0,2)	0-(2,1) 1-(3,2)	0-(3,3) 0-(4,0)	0-(2,0) 1-(1,0)	0-(2,0) 1-(1,1)

Table A.2.9: Parameters of the estimated models based on 200 data points of the first 5 data sets of the second switch

Method \ Test set	1	2	3	4	5
1-step forecasts					
ARIMA	1.4	18.6	4.2	5.9	3
VARMA	7.4	**2.1**	**1**	11	**0.4**
VARMAX	2	24.8	19.5	**1.2**	13.7
Linear regression with ARMA errors	**0.6**	212.8	8.7	3.7	0.8
Linear regression	3.9	159.3	84.1	3.2	15.1
10-step forecasts					
ARIMA	26.5	251.8	73.4	60	49.1
VARMA	**15.9**	385.6	**28.3**	65.2	**31**
VARMAX	29.5	**32.8**	33.2	**9.6**	71.2
Linear regression with ARMA errors	53.5	133.1	72.2	20.9	48.4
Linear regression	36.9	38.6	109.3	12.4	66.1
50-step forecasts					
ARIMA	99.4	122.7	205	76.5	108.2
VARMA	90.2	214.1	66.3	75.2	**68.2**
VARMAX	**31**	**67.7**	**35**	69.4	76.3
Liner regression with ARMA errors	40	185.1	45.5	46.9	68.3
Linear regression	36.2	44.9	51.2	**46.2**	73.7

Table A.2.10: Forecasts based on a training set of 200 data points for the first 5 data sets of the second switch

Method \ Test set	6	7	8	9	10
ARIMA	(4,1,1)	(3,1,4)	(2,1,1)	(3,0,3)	(3,1,3)
VARMA	(1,0)	(1,1)	(1,0)	(4,0)	(1,0)
VARMAX	(1,0)	(1,1)	(1,0)	(1,0)	(1,0)
Liner regression with ARMA errors	0-(4,2) 1-(2,0)	0-(1,0) 1-(1,0)	0-(3,2) 0-(1,0)	0-(1,0) 1-(4,4)	0-(4,1) 1-(3,2)

Table A.2.11: Parameters of the estimated models based on 200 data points of the last 5 data sets of the second switch

Method \ Test set	6	7	8	9	10
1-step forecasts					
ARIMA	2.2	**65.5**	1.7	**6.1**	0.3
VARMA	4.2	88.1	6.9	26.7	**0**
VARMAX	**0.2**	139.5	**0.1**	15.7	1.3
Linear regression with ARMA errors	4.6	102	3.3	77.3	1.8
Linear regression	52.9	219.7	8.4	31.3	17.3
10-step forecasts					
ARIMA	88.1	52.9	40.5	107.3	**4.6**
VARMA	**79.9**	37.3	33.8	143.8	9
VARMAX	183.7	20.1	**18.3**	**29**	34.8
Linear regression with ARMA errors	129.5	**15.9**	24.3	42.4	32.6
Linear regression	168.2	33.4	25.7	32.8	38.9
50-step forecasts					
ARIMA	300	121.1	103.5	162.7	158.6
VARMA	**72.4**	101.1	119.4	171.5	**37.2**
VARMAX	180.5	**40.7**	64.7	16.4	59.9
Liner regression with ARMA errors	130.4	43.7	65.9	35.8	58.2
Linear regression	137.1	46.4	**43.8**	**30.2**	73

Table A.2.12: Forecasts based on a training set of 200 data points of the last 5 data sets of the second switch

A.3 Forecasts with other sample sizes and combinations of Switch 3

A.3.1 Forecasts based on a training set of 50 data points

Test set / Variable(s)	1	2	3	4	5	forecasts combined
MeasurementTemperature	82	138.8	21.9	71	49.1	72.6
T Bottom (degC)	137.4	103.2	64.9	213.7	109.7	125.8
T LB HOT (degC)	91.5	87.6	77.8	113.7	72.6	88.6
T LB COLD (degC)	109.7	108-8	135.8	137.9	81.5	114.7
Temp outdoor (deg C)	87.7	88.6	36.4	120.7	53.5	77.4
Temp inside (deg C)	131.7	122	110.4	152	94.8	122.2
T LB HOT (degC), Temp outdoor (degC) (best temperature combination)	116.3	88.4	82.2	168	68.2	104.6
T Bottom (degC), T LB HOT (degC)	**45.1**	55.8	54.5	65.7	33.7	51
All measured temperatures except T Bottom (degC)	108	106.4	206.2	95.4	44.1	112
All 6 measured temperatures	106.2	65.7	175	62.3	61.2	94.1
T LB COLD (degC), Temp outdoor (deg C), Temp inside (deg C) (worst temperature combination)	86.3	123.7	347.6	203.9	82.8	168.9
Humidity outdoor (%RH)	101.4	89.9	31.8	98.8	62.7	76.9
Humidity inside (%RH)	90.6	55	29	57.3	48.1	56
Humidity outdoor (%RH), Humidity inside (%RH)	121	52.1	21.1	51.6	49	59
T LB HOT (degC), Humidity inside (%RH)	97.1	85.4	19.6	85.3	50.3	67.5
T Bottom (degC), T LB COLD (degC), Temp outdoor (deg C), Humidity outdoor (%RH), Humidity inside (%RH)	91.9	20	**14.3**	63	27	43.2
T LB COLD (degC), Temp outdoor (deg C), Humidity outdoor (%RH), Temp inside (deg C) (worst combination among all variables)	91.5	118.8	396.8	190.4	106.3	180.8
T Bottom (degC), Humidity inside (%RH) (best combination among all variables)	46.3	**10.7**	31.1	**32.8**	**17.8**	**27.7**

Table A.3.1: Forecasts based on a training set of 50 data points of the third switch

A.3.2 Forecasts based on a training set of 150 data points

Variable(s) \ Test set	1	2	3	4	5	forecasts combined
MeasurementTemperature	55.3	155.3	30.6	64.3	57.9	72.7
T Bottom (degC) (worst combination among all variables)	108.4	67.8	74.4	109.1	56.5	83.2
T LB HOT (degC)	49.6	51.1	24.9	42.9	36.4	41
T LB COLD (degC)	49	63.1	34.4	33.5	42.4	44.5
Temp outdoor (deg C)	31.7	77.8	39.2	80.2	37.9	53.4
Temp inside (deg C)	67.1	69.4	33.2	44.9	46.8	52.3
T LB HOT (degC), Temp outdoor (degC)	49.6	51.1	24.9	42.9	36.4	41
T Bottom (degC), T LB HOT (degC) (best temperature combination)	37.5	**24.2**	36.1	28.6	16.2	28.5
All measured temperatures except T Bottom (degC)	25.6	63.7	30.7	66.9	37.2	44.8
All 6 measured temperatures	48.3	32.3	46	38.8	24	37.9
Humidity outdoor (%RH)	52.2	77.8	30.3	30	57	49.4
Humidity inside (%RH)	65.1	43.8	23.7	12.7	34.1	35.9
Humidity outdoor (%RH), Humidity inside (%RH)	59	64	28.5	**11.4**	27.6	38.1
T LB HOT (degC), Humidity inside (%RH)	98	46.5	22	13.5	39	43.8
T Bottom (degC), T LB COLD (degC), Temp outdoor (deg C), Humidity outdoor (%RH), Humidity inside (%RH)	27.3	32.6	14	12.3	**14.1**	20.1
MeasurementTemperature, T Bottom (degC), T LB COLD (degC), Temp outdoor (deg C), Humidity outdoor (%RH), Humidity inside (%RH) (best combination among all variables)	**26.5**	24.8	**13.8**	13.3	14.6	**18.6**

Table A.3.2: Forecasts based on a training set of 150 data points of the third switch

A.3.3 Forecasts based on a training set of 200 data points

Variable(s) \ Test set	1	2	3	4	5	forecasts combined
MeasurementTemperature	63.2	154.3	26.5	75.3	66.5	77.2
T Bottom (degC) (worst combination among all variables)	115.1	80.6	51.4	111.7	76.1	87
T LB HOT (degC)	63.6	50.2	23.8	45.6	43	45.2
T LB COLD (degC)	62.3	62.5	32.9	35.1	50.3	48.6
Temp outdoor (deg C)	32.2	78.2	43.9	85.6	41.9	56.4
Temp inside (deg C)	81.8	68.4	29.5	47.8	55.2	56.6
T LB HOT (degC), Temp outdoor (degC)	33	74.2	32.1	49.5	45.2	46.8
T Bottom (degC), T LB HOT (degC)	55.4	**28.3**	33.5	32.4	18.3	33.6
All measured temperatures except T Bottom (degC)	27.5	58.1	49.2	61.1	37	46.6
All 6 measured temperatures	73.1	33.6	45.6	39.2	15.7	41.4
T Bottom (degC), T LB COLD (degC), Temp outdoor (deg C) (best temperature combination)	30.9	36	49.7	28.9	13.9	31.9
Humidity outdoor (%RH)	109.3	78.7	27	26.6	65.1	61.4
Humidity inside (%RH)	145.2	45.8	19.8	11.6	41	52.7
Humidity outdoor (%RH), Humidity inside (%RH)	115.5	62.4	19.8	10.6	37.2	49.1
T LB HOT (degC), Humidity inside (%RH)	57.7	45.6	**17**	14.6	41.3	35.2
T Bottom (degC), T LB COLD (degC), Temp outdoor (deg C), Humidity outdoor (%RH), Humidity inside (%RH)	28.5	31.6	69.5	13.5	**12**	31
MeasurementTemperature, T Bottom (degC), T LB HOT (degC), Temp outdoor (deg C), Humidity inside (%RH) (best combination among all variables)	**20.8**	31	29.6	**10.4**	12.9	**21**

Table A.3.3: Forecasts based on a training set of 200 data points of the third switch

Bibliography

[1] Thomas Böhm. “Präzise Prognose von Störungen an Eisenbahnweichen (Precise Prediction of Failures of Railway Switches)”. PhD thesis. Feb. 2018.

[2] Bernhard Bollrath and Uwe Sattler. “Weichendiagnosesystem Sidis W – Betriebliche Erfahrungen”. In: *ETR – Eisenbahntechnische Rundschau, Nr. 10* (2006).

[3] George EP Box et al. *Time series analysis: forecasting and control*. Fifth edition. John Wiley & Sons, 2015.

[4] Peter J. Brockwell and Richard A Davis. *Time Series: Theory and Methods*. Second edition. Springer-Verlag, 1991.

[5] Patrizia Campagnoli, Sonia Petrone, and Giovanni Petris. *Dynamic Linear Models with R*. First edition. Springer, 2009.

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

[7] David A. Dickey and Wayne A. Fuller. “Distribution of the Estimators for Autoregressive Time Series With a Unit Root”. In: *Journal of the American Statistical Association* 74.366 (1979), pp. 427–431.

[8] Randal Douc, Eric Moulines, and David Stoffer. *Nonlinear time series: Theory, methods and applications with R examples*. Chapman and Hall/CRC, 2014.

[9] James Durbin and Siem Jan Koopman. *Time Series Analysis by State Space Methods*. Second edition. Oxford university press, 2012.

[10] Beate Dutschk et al. “Deeper insight in railway switch condition nowcasting”. In: *WCCM 2017 - 1st World Congress on Condition Monitoring*. British Institute of Non-Destructive Testing, 2017.

[11] Walter Enders. *Applied Econometric Time Series*. First edition. Wiley series in probability and mathematical statistics. Wiley, 1995.

[12] Daniel Fässler, Daniela Narezo Guzman, and Thorsten Neumann. “Modellierung der Stromaufnahme von Weichenantrieben”. In: *EI - Der Eisenbahningenieur* 5/2020 (2020), pp. 29–33.

[13] Wayne A. Fuller. *Introduction to Statistical Time Series*. Second edition. John Wiley & Sons, 2009.

[14] Fausto Pedro García Márquez, Diego Pedregal, and Clive Roberts. “Time series methods applied to failure prediction and detection”. In: *Reliability Engineering & System Safety - RELIAB ENG SYST SAFETY* 95 (June 2010), pp. 698–703.

[15] Daniela Narezo Guzman, Thorsten Neumann, and Jörn Christoffer Groos. “Kontinuierliche Überwachung der LST mit eingebetteten Sensoren”. In: *Der Eisenbahningenieur EI* (2018), pp. 6–11.

[16] Daniela Narezo Guzman et al. “Data-driven condition now- and forecasting of railway switches for improvement in the quality of railway transportation”. In: *4th European Conference of the Prognostics and Health Management (PHM) Society*. Ed. by Chetan S. Kulkarni and Tiedo Tinga. Proceedings of the European Conference of the PHM Society, 2018.

[17] M T Hagan and S M Behr. “The time series approach to short term load forecasting”. In: *IEEE Trans. Power Syst.; (United States)* (Aug. 1987).

[18] James D. Hamilton. *Time Series Analysis*. First edition. Princeton University Press, 1994.

[19] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

[20] K. Liu et al. “Comparison of very short-term load forecasting techniques”. In: *IEEE Transactions on Power Systems* 11.2 (1996), pp. 877–882.

[21] Helmut Lütkepohl. *New Introduction to Multiple Time Series Analysis*. Springer Science & Business Media, 2005.

[22] Shamsul Masum, Ying Liu, and John Chiverton. “Multi-step Time Series Forecasting of Electric Load Using Machine Learning Models”. In: 2018, pp. 148–159.

[23] Thorsten Neumann and Daniela Narezo Guzman. “Bayesian network design for fault diagnostics of railway switches”. In: *29th European Safety and Reliability Conference (ESREL 2019)*. Research Publishing Services, 2019, pp. 1117–1124.

[24] Thorsten Neumann, Daniela Narezo Guzman, and Jörn Christoffer Groos. “Transparente Fehlerdiagnose bei Weichenstörungen mittels Bayes’scher Netze”. In: *SIGNAL + DRAHT* (2019), pp. 23–31.

[25] Klaus Neusser. *Zeitreihenanalyse in den Wirtschaftswissenschaften*. Third edition. Springer, 2011.

[26] Stylianos Pappas et al. “Electricity Demand Loads Modeling using AutoRegressive Moving Average (ARMA) models”. In: *Energy* 33 (Sept. 2008), pp. 1353–1360.

[27] Fabian Pedregosa et al. “Scikit-learn: Machine learning in Python”. In: *Journal of machine learning research* 12.Oct (2011), pp. 2825–2830.

[28] Rob Redeker. “POSS®: Railway condition monitoring developed by a maintainer”. In: *5th IET Conference on Railway Condition Monitoring and Non-Destructive Testing*. 29.-30. Nov. 2011, Derby, UK.

[29] Skipper Seabold and Josef Perktold. “statsmodels: Econometric and statistical modeling with python”. In: *9th Python in Science Conference*. 2010.

[30] Pauli Virtanen et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: *Nature Methods* 17 (2020), pp. 261–272.

Name: Daniel Fässler Matrikelnummer: 746472

Erklärung

Ich erkläre, dass ich die Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Ulm, den .....

Daniel Fässler