



TECHNISCHE
UNIVERSITÄT
WIEN

Vienna University of Technology

DIPLOMARBEIT

Prognose von Spotpreisen für Energie mit Support Vector Regression

Ausgeführt am Institut für
Wirtschaftsmathematik
der Technischen Universität Wien

unter der Anleitung von

Ao.Univ.Prof. Wolfgang Scherrer

durch

Tina Laaber

Schrottgasse 11/16
1030 Wien

Datum

Unterschrift

Ich danke Ao.Univ.Prof. Wolfgang Scherrer für die intensive Betreuung dieser Diplomarbeit sowie DI Dr. Gerold Petritsch und DI Katharina Vedovelli von der e&t Energie Handelsgesellschaft m. b. H. für die Bereitstellung der Daten und Hilfestellungen. Des weiteren möchte ich den beiden Korrekturlesern Wolfgang Eichert und Alfred Pabst sowie meiner Familie und meinen Freunden für die Unterstützung während meiner gesamten Studienzzeit danken.

Inhaltsverzeichnis

Einleitung	1
1 Der Handel von Spotkontrakten auf Strom an der European Energie Exchange	1
1.1 Allgemeines	1
1.2 Ablauf und Produkte am Spotmarkt	2
1.2.1 Der tägliche Ablauf des Handels	2
1.2.2 Auftragsarten für Stundenkontrakte auf Strom	2
1.2.3 Auftragsarten für Blockkontrakte auf Strom	3
1.3 Preisbildung am Stromspotmarkt	4
1.4 Extreme Werte in den Strompreisen	5
1.5 Negative Preise	6
2 Zeitreihenanalyse: Stromspotpreise an der EEX	6
2.1 Zeitreihen	7
2.2 Trendanalyse	8
2.3 Autokorrelationsfunktion und partielle Autokorrelationsfunktion	9
2.4 Periodogramm	12
2.5 Gleitende Mittelwerte und Varianzen	15
3 Exogene Variablen	18
3.1 Wind	18
3.2 Temperatur	18
3.3 Lastprognosen	18
3.4 Strombörse EXAA	19
4 Prognose: Einleitung und theoretische Grundlagen	19
4.1 Prognosemodelle	19
4.1.1 Einfaches lineares Modell ohne exogene Variablen	19
4.1.2 Einfaches lineares Modell mit exogenen Variablen	20
4.1.3 Großes Modell ohne exogene Variablen	21
4.1.4 Großes Modell mit exogenen Variablen	21
4.2 Prognoseverfahren	22
4.2.1 Naive Prognose	22
4.2.2 Die Methode der kleinsten Quadrate	23
4.2.3 Die Methode der kleinsten Quadrate mit Gewichten	23
4.2.4 Support Vector Regression	24
4.2.5 Support Vector Regression mit Gewichten	26
4.3 Qualitätskriterien	27
4.3.1 Mittlere Absolute Abweichung	27
4.3.2 Mittlere quadratische Abweichung	27
4.3.3 Hitrate	27
4.3.4 Autokorrelation der Fehler	28
4.4 AIC Schleife	28

5	Prognose: Praktische Umsetzung	29
5.1	Verwendete Bibliotheken	29
5.2	Selbst geschriebene Programme	30
5.3	Naive Prognose	31
5.4	Kleinste Quadrate Prognose	31
5.5	Kleinste Quadrate Prognose mit Gewichten	36
5.6	Support Vector Regression	38
5.6.1	Parameter der Kernels	38
5.6.2	Parameter der eps- und nu-regression	40
5.6.3	Ergebnis	42
5.7	Support Vector Regression mit Gewichten	43
5.8	Vergleich der Ergebnisse	44
5.9	Veränderung des Ergebnis durch Verwendung des AIC Kriteriums	46
5.10	Einfluss der einzelnen exogenen Variablen	47
6	Conclusio	48
	Appendix	50
	Literatur	50
	Abbildungsverzeichnis	51
	Tabellenverzeichnis	53
A	Quellcodes	54
A.1	Daten einlesen	54
A.2	Zeitreihenanalyse	55
A.3	Modelle bauen	60
A.4	AIC Algorithmus	65
A.5	Naive Prognose	67
A.6	Kleinste Quadrate Prognose	68
A.7	Verallgemeinerte kleinste Quadrate Prognose	72
A.8	Support Vector Regression mit Gewichten	78
A.9	Support Vector Regression	80

Es soll festgehalten werden, dass die Verwendung männlicher Formen in dieser Diplomarbeit ausschließlich wegen der leichteren Lesbarkeit vorgenommen wurde und als auf Frauen und Männer bezogen verstanden werden soll.

Einleitung

In dieser Arbeit geht es um die Prognose von Stromspotpreisen an der European Energy Exchange, die im Allgemeinen eine sehr komplexe und nicht leicht prognostizierbare Struktur haben. Die Datengrundlage wurde dabei in Zusammenarbeit mit der e&t Energie Handelsgesellschaft m.b.H.(e&t) zusammengestellt und beinhaltet die stündlichen Strompreise sowie verschiedene exogene Variablen. Ziel dieser Arbeit war es einerseits, herauszufiltern in wie weit die gewählten Parameter einen Einfluss auf den Strompreis haben und andererseits, naive Prognose, Prognose mittels kleinster Quadrate Schätzung und Support Vector Regression zu vergleichen. Im weiteren Verlauf erwies sich zusätzlich als Verbesserung noch ein verallgemeinertes Regressionsmodell zu verwenden sowie diese Verallgemeinerung auf die Support Vector Regression zu übertragen. Diese Arbeit ist in sechs Teile gegliedert. Als erstes wird der Spotmarkt für Strompreise vorgestellt, weiters wird die Zeitreihe der stündlichen Preise untersucht um die Struktur der Daten zu erfassen. Im dritten Kapitel werden dann die exogenen Variablen erläutert, im vierten Kapitel werden die Prognoseansätze vorgestellt sowie auch die theoretischen Grundlagen erläutert. Im darauffolgenden Kapitel werden die praktische Umsetzung sowie auch die Ergebnisse erläutert. Abschließende Schlussfolgerungen befinden sich im letzten Kapitel.

1 Der Handel von Spotkontrakten auf Strom an der European Energie Exchange

Alle Informationen dieses Kapitels stammen aus verschiedenen Dokumenten und Texten von der European Energy Exchange (EEX) Homepage und der European Power Exchange (EPEX) Homepage. Genauere Angaben dazu befinden sich im Literaturverzeichnis unter den Referenzen [EEX1] bis [EEX3].

1.1 Allgemeines

Die EEX entstand 2002 durch den Zusammenschluss der Frankfurter und Leipziger Strombörse. Zusammen mit der französischen Powernext SA hat die EEX eine Tochtergesellschaft, die EPEX, an der vor allem der Energiespotmarkt durchgeführt wird. Spotmarkt bezieht sich auf den kurzfristigen Handel und die EPEX handelt den Strom für die Märkte Frankreich, Schweiz, Deutschland und Österreich wobei Österreich und Deutschland zu einem Marktgebiet zusammengeschlossen sind. Dabei wird der Strom für den darauffolgenden Tag (Day Ahead) im Auktionshandel, sowie kontinuierlich für denselben Tag, gehandelt. Der Spotpreis für das Marktgebiet Österreich/Deutschland gilt als europaweiter Referenzwert. Insgesamt wurden 2009 203 TWh in den drei Marktgebieten gehandelt. In den folgenden Abschnitten bezeichnen sowohl EEX als auch die Tochtergesellschaft EPEX die Börsen für den Spotmarkt für Strom.

1.2 Ablauf und Produkte am Spotmarkt

Prinzipiell werden am EEX Spotmarkt Strom und Emissionsrechte gehandelt. Hierbei gibt es verschiedene Arten von Handel, Auktionen und fortlaufender Handel. Weiters wird zwischen Stundenkontrakten, also Handel von einer festgelegten Stunde, und Blockkontrakten, also Handel zu mehreren Stunden zu einem Block zusammengefasst, unterschieden. Einzelne Stunden werden in Auktionen abgewickelt, Emissionsrechte und Blockkontrakte werden in Auktionen und fortlaufend gehandelt. Es werden an jedem Börsetag für den Folgetag die Auktionen durchgeführt. Jeder Börsenteilnehmer, das entspricht einem zum Handel zugelassenes Unternehmen, beteiligt sich mittels Händlern am Kauf und Verkauf. Dabei wird unterschieden, wer das Recht auf Kauf- und Verkaufsangebote hat, wer Strom und wer Emissionsrechte handeln darf. Bei Blockkontrakten erfolgt der Handel individuell pro Händler, bei Stundenkontrakten werden die Angebote eines Handelskontos zusammengeschlossen. Ein Börseteilnehmer hat einen oder mehrerer Händler, aber -außer explizit gewünscht- nur ein Handelskonto. Des weiteren gibt es für jeden Börseteilnehmer Regelzonen, in denen er am Handel teilnehmen darf, welche später noch genauer beschrieben werden. In den folgenden Punkten dieses, sowie den folgenden Abschnitten werden nur die für diese Arbeit relevanten Stundenkontrakte auf Strom im Marktgebiet Österreich/Deutschland besprochen.

1.2.1 Der tägliche Ablauf des Handels

Es gibt drei Handelsphasen: den Vorhandel, den Haupthandel und den Nachhandel. Die Börse ist für ihre Teilnehmer täglich von 7:30 bis 20:00 zugänglich. In der Vorhandelsphase darf jeder Börsenteilnehmer bis zu 14 Tage im vorhinein für seine Regelzonen Kauf bzw. Verkaufsangebote legen und diese modifizieren. Diese Phase ist bis am Vortag der Lieferung, dem Handelstag, bis am späten Vormittag offen. Die genaue Uhrzeit richtet sich nach dem Start der Preiskalkulation, welche sich wieder nach dem jeweiligen Marktgebiet unterscheidet. Für alle gilt aber, dass ab Beginn der Preiskalkulation keine Änderungen mehr durchgeführt werden können. Die Preisermittlung für Österreich und Deutschland startet um 12:00 mit der Schließung des Orderbuches. In der Haupthandelsphase werden die Preise für alle Stunden je Marktgebiet ermittelt und bei Erfolg werden diese in der anschließenden Preisveröffentlichung, der Nachhandelsphase bekanntgegeben. Für das Marktgebiet Österreich/Deutschland ist dies ab 12:15. Weiters werden noch weitere Information wie Base- und Peakloadpreise sowie Handelsumsatz veröffentlicht. Falls keine Preise für alle Stundenkontrakte gefunden werden, so kann die EPEX alle oder ausgewählte Stunden erneut zur Auktion freigeben.

1.2.2 Auftragsarten für Stundenkontrakte auf Strom

Ein Stundenkontrakt bezieht sich immer genau auf eine ausgewählte Stunde, einen bestimmten Liefertag und einen Lieferort. Jeder Börsenteilnehmer ist für gewisse Lieferorte freigestellt. Diese sind für das Marktgebiet Österreich/Deutschland die Regelzonen der Amprion GmbH, Transpower Stromübertragungs GmbH, Vattenfall Europe Transmission GmbH, ENBW Transportnetz AG und Verbund-Austrian Power Grid AG. Es gilt immer das letzte

Angebot, das pro Börsenteilnehmer abgegeben wurde; es kann also ein Händler das Angebot eines anderen Händlers ungültig machen, indem er für den selben Lieferort, Tag und Stunde noch ein weiteres Angebot legt. Ein Stundenangebot ist eine stückweise lineare Funktion zwischen minimalem und maximalem Preis. Es wird also mittels Wertepaaren genau festgelegt, wieviel bei einem gegebenen Strompreis gekauft bzw verkauft wird. Die Preise sind in €/MWh und die Mengen in MWh angegeben und Mengen sind positiv für Käufe, negativ für Verkäufe. Zwischen zwei Wertepaaren wird linear interpoliert und es müssen alle Werte zwischen einem minimalen Preis von -3000€/MWh und maximalen Wert von 3000€/MWh festgelegt sein. Ein Beispiel für die Wertepaare befindet sich in Tabelle 1 und die dazugehörige Grafik in Abbildung 1. Die Funktion muss monoton sein und es sind auch preisunabhängige Angebote möglich. Das heißt konkret, dass für ein Verkaufsangebot eine höhere Menge immer einen höheren Preis hat und für ein Kaufangebot eine höherer Menge immer einen niedrigeren Preis hat.

Preis in €/MWh	-3000	0	40	50	1000	3000
Menge in MWh	500	500	400	0	-300	-300

Tabelle 1: Wertepaare

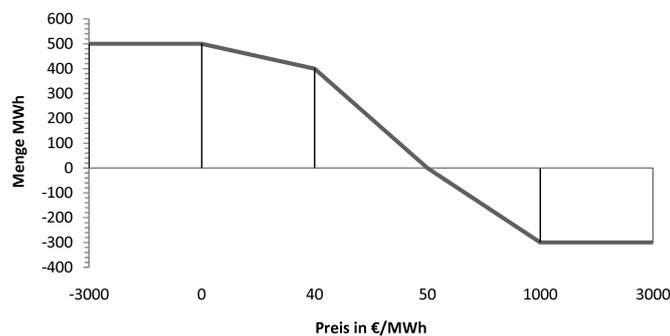


Abbildung 1: Beispiel eines Stundenkontrakts

1.2.3 Auftragsarten für Blockkontrakte auf Strom

Nachdem die Blockkontrakte einen Einfluss auf den Preis der Stundenkontrakte haben, werden diese hier kurz besprochen. Ein Blockkontrakt sind mehrere Stundenkontrakte, also Teilaufträge die zu einem Block also Gesamtauftrag zusammengefasst werden. Blockkontrakte erfüllen folgende Bedingungen:

- Ein Block umfasst zwischen 2 und 24 Stundenkontrakten
- Sie müssen den gleichen Liefertag und Ort besitzen
- Sie müssen auf die gleiche Strommenge lauten und zwischen 0 MWh und 300 MWh für das Marktgebiet Österreich/Deutschland liegen

- Pro Börsenteilnehmer sind maximal 45 Blockangebote möglich (außer er besitzt mehrere Handelskonten)

Wenn ein Blockangebot über ein Preislimit verfügt, dann wird es z.B. für einen Kauf ausgeführt wenn der Mittelwert der betreffenden Stunden unter dem Limitpreis liegt oder gleich dem Limitpreis ist. Zusätzlich gibt es auch unlimitierte Angebote, diese liegen aber immer zwischen dem minimalen Preis von -3000€/MWh und maximalen Preis von 3000€/MWh.

1.3 Preisbildung am Stromspotmarkt

Für die Preisermittlung werden Stunden und Blockkontrakte zusammengeführt und eine gemeinsame Auktion ausgeführt. Dabei werden die Blockkontrakte in preisunabhängige Stundenkontrakte umgeändert. Weiters werden die verschiedenen Lieferorte, wo ein Austausch ohne Lieferengpässe möglich ist (dies ist im Allgemeinen der Fall, für die Sonderregelungen im Falle einer Aufteilung des Marktgebietes wird auf [EEX3] verwiesen), zu einem Marktgebiet zusammengefasst. Insgesamt hat also eine Stunde, an einem Tag, in einem Marktgebiet genau einen Preis. Im deutschsprachigen Raum gibt es zwei Marktgebiete, Schweiz und Österreich/Deutschland. Durch die Blockkontrakte sind die Stunden wechselseitig voneinander abhängig und die Preisermittlung kann nur für alle 24h in einer gemeinsamen Auktion ermittelt werden. Die Preisbildung erfolgt über einen Iterationsprozess, der solange durchgeführt wird, bis entweder alle Stunden eines Blockangebotes im Preislimit liegen, oder keine Stunde darin liegt. Dazu werden zunächst 24 einzelne Preise für jede Stunde ermittelt und im nächsten Schritt wird für jene Stunden, in denen nicht alle Kontrakte erfüllt sind wieder ein neuer Preis ermittelt. Dies wird solange gemacht bis der Iterationsprozess zu Konvergenz führt, also alle Kontrakte innerhalb der Limits liegen. Der Iterationsprozess läuft folgendermaßen ab:

Es werden alle Stunden und Blockkontrakte für eine fixe Stunde zu einer aggregierten Angebots- bzw. Nachfragekurve zusammengeführt. Dazu werden zu jedem Preis zwischen Minimum und Maximum die Mengen zusammengezählt und der Schnittpunkt bestimmt das Gleichgewicht zwischen Käufen und Verkäufen. Ein Beispiel dafür befindet sich in Abbildung 2. Als nächster Schritt, sofern nötig, wird ein Blockangebot aus der Preisermittlung herausgenommen. Das ausgewählte Blockangebot ist jenes mit der größten Limitüberschreitung (dazu wird der Durchschnittspreis betrachtet) aus den verbliebenen Blockangeboten. Nach Konvergenz des Verfahrens werden Mithilfe der resultierenden Preise die Mengen festgelegt und schließlich beides veröffentlicht.

Wenn kein Preis über die Angebots und Nachfragekurve ermittelt werden kann, beziehungsweise wenn der Preis außerhalb der minimalen und maximalen Preise liegt, werden die Gründe dafür veröffentlicht und eine weitere Auktion findet statt.

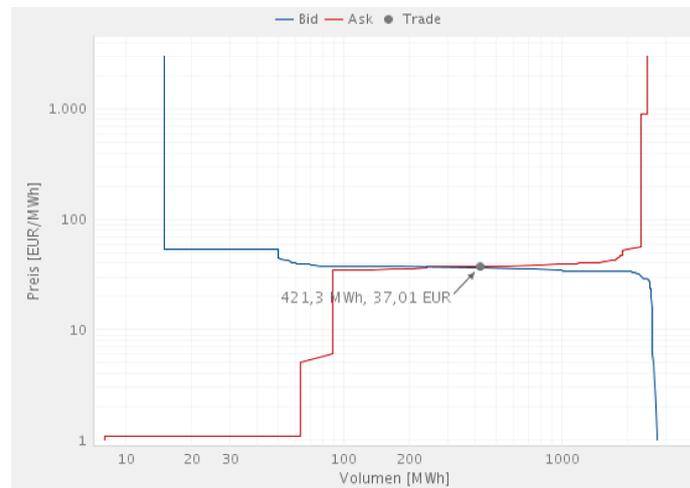


Abbildung 2: Beispiel einer aggregierten Angebots- bzw. Nachfragekurve an der EXAA 25. März 2010 Stunde 18

(Quelle: www.exaa.at [EXAA1])

1.4 Extreme Werte in den Strompreisen

Das Problem bei diesen Werten liegt dabei sie erstens zu identifizieren, zweitens im Modell zu berücksichtigen und drittens die Prognose nicht zu verfälschen. Es ist schwierig zu definieren, ab welcher Schwelle ein Wert als Ausreißer betrachtet werden kann. Dieser Punkt wird in dieser Arbeit aber nicht weiter diskutiert. Im Modell kommt das Problem daher, dass ein extremer Wert bei der Modellierung auch dargestellt werden sollte. Andererseits ist trotz mathematischer Modellansätze die genaue Betrachtung des momentanen Umfeldes unerlässlich. Die Frage stellt sich, in wie weit die Information für extreme Werte schon bekannt ist. Zum Beispiel kann der Ausfall eines Kernkraftwerkes, das einen großen Beitrag zum Gesamtvolumen leistet, den Preis in die Höhe treiben. Diese Information ist für alle Marktteilnehmer zugänglich und sollte abgesehen der Prognosemodelle auch berücksichtigt werden.

Um die Anzahl der Ausreißer zu quantifizieren wurden in der Datenreihe die Werte betrachtet die absolut über $150\text{€}/\text{MWh}$ liegen. Deren gibt es 91 und dabei fiel auf, dass die Werte teilweise einzeln und teilweise in Gruppen auftreten wobei bei zweitem das Phänomen bereits nach 3-4 Stunden wieder abklingt. Weiters wurde noch betrachtet ob die Ausreißer zu bestimmten Stunden bzw. Tageszeiten vermehrt auftreten. Die Ergebnisse befinden sich in Grafik 3 und man sieht, dass die Ausreißer vor allem in den Mittags- und Abendstunden auftreten.

Wenn man die Ausreißer ohne Modifikation in der Zeitreihe verwendet führt dies allerdings zu drei Problemen. Erstens wird der Ausreißer durch diesen vereinfachten Ansatz nicht richtig prognostiziert und führt damit zu einem großen Prognosefehler und zweitens wird der extreme Wert wieder für weitere Prognosen verwendet und verfälscht damit die darauffolgenden Werte, weiters versucht der Parameterschätzer (der kleinste Quadrate Schätzer reagiert

z.B. sehr sensibel auf Ausreißer) den Ausreißer möglichst gut einzubeziehen und verfälscht damit auch die Prognose.

In der nachfolgenden Prognose wird das Problem der Ausreißer zum Teil in der Support Vector Regression berücksichtigt. Bei dieser Methode werden Ausreißer mit einem linearen Strafterm bestraft. Genauere Erklärungen dazu folgen noch im weiteren Verlauf der Arbeit.

1.5 Negative Preise

Seit kurzem sind auch negative Strompreise an der EEX möglich; die Abnehmer bekommen also Geld dafür, dass sie den Strom nehmen. Bei normalen Schätzverfahren führt dies zu keinen weiteren Problemen. Wenn man sich aber dafür entscheidet, mit logarithmierten Werten zu rechnen ist ein Behandeln der negativen Werte unerlässlich. Eine Möglichkeit wäre diese aus der Zeitreihe zu entfernen, wodurch am entsprechenden Platz dann Werte für einzelne Stunden fehlen würden. In dieser Arbeit wird nur mit nicht logarithmierten Werten gerechnet und daher stellen die negativen Strompreise kein Problem dar. Um auch wie schon bei den Ausreißern ein Bild über die negativen Preise zu geben wurde die Zeitreihe näher betrachtet. Es gibt in der betrachteten Zeitreihe 69 negative Werte und dabei fiel auf, dass die negativen Werte einzeln bzw. in Gruppen von 3-5 Stunden auftreten. Also etwas mehr als die Ausreißer zu einem Clustering neigen. Bei näherer Betrachtung der Verteilung über die einzelnen Stunden fällt auf, dass negative Preise vor allem in den Morgenstunden auftreten (siehe auch Abbildung 3).

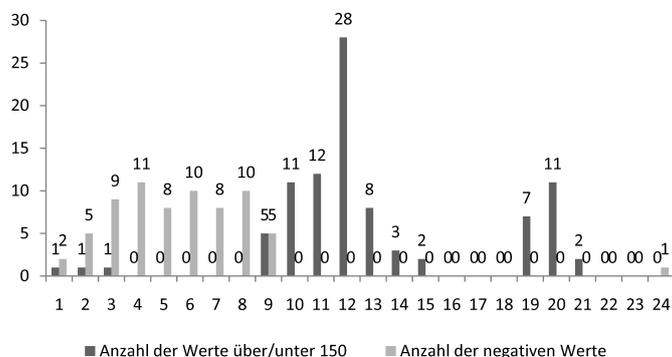


Abbildung 3: Ausreißer und negative Werte

2 Zeitreihenanalyse: Stromspotpreise an der EEX

Als erster Schritt für eine Prognose ist die ausführliche Analyse der Daten unerlässlich. Grundlage für die Analyse waren die stündlichen Spotpreise der EEX von 1 Jänner 2008 bis 15 Dezember 2009. Die Strompreise wurden in verschiedene Datenreihen zerlegt. Zunächst wurde die gesamte Zeitreihe betrachtet und in einem weiteren Schritt die Zeitreihen der einzelnen Stunden sowie

die Zeitreihen der mittleren, maximalen und minimalen Preise eines jeden Tages. Die Zeitreihe der Mittelwerte erwies sich als äußerst nützlich in der Prognose als geglättete Basisreihe für den Vergleich mehrerer Methoden. Die minimalen und maximalen Tagespreise wurden lediglich interessehalber dazugekommen. In der Analyse wurden Trends, Saisonalitäten, gleitende Mittelwerte und Varianzen, Autokorrelationsfunktionen, partielle Autokorrelationsfunktionen betrachtet sowie tagesspezifische Mittelwerte und Varianzen betrachtet. Ziel der Zeitreihenanalyse ist es, das Verhalten bis auf ein weißes Rauschen möglichst gut erklären zu können. Theoretische Grundlagen der Zeitreihenanalyse kommen aus Time Series: Theory and Methods [BD] sowie aus Econometrics II [ÖK2]. Alle Schritte der Zeitreihenanalyse werden in Vorbereitung für die spätere Formulierung des Prognosemodells gemacht.

2.1 Zeitreihen

Als erster Schritt jeder Analyse sollte die Zeitreihe graphisch dargestellt werden. Dies dient dazu ein erstes Gefühl für eventuelle Trends, Saisonalitäten, Strukturbrüche usw. zu bekommen. Diese Vermutungen müssen in weiterer Folge mit Methoden der Zeitreihenanalyse untersucht werden. Graphisch dargestellt wurde die gesamte Zeitreihe (Abbildung 4), sowie exemplarisch einige Stunden (Abbildung 5), die mittleren, minimalen und maximalen Tagespreise (Abbildung 6). Um die Grafik übersichtlicher zu gestalten ist hier immer nur ein Ausschnitt der Zeitreihe dargestellt.

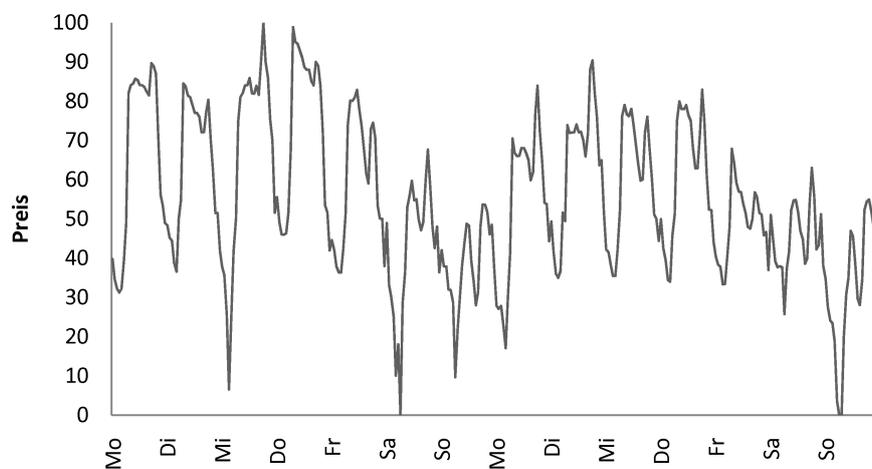


Abbildung 4: Gesamte Zeitreihe von 14.1.2008 bis 27.1.2008

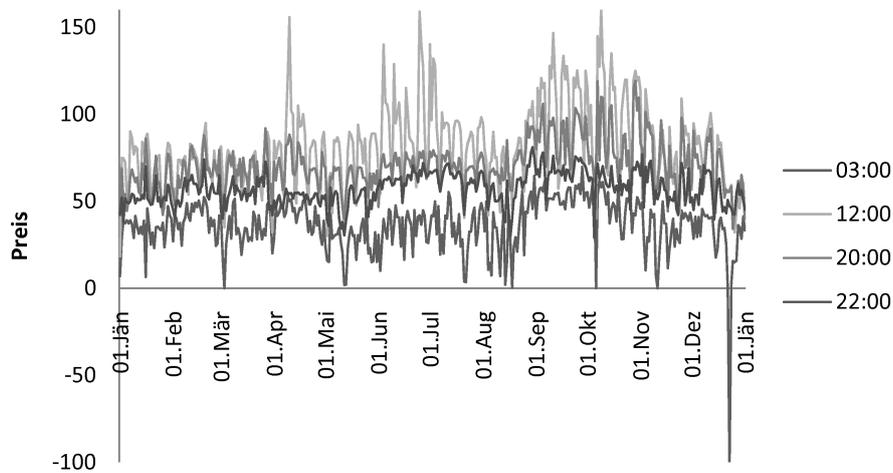


Abbildung 5: Zeitreihe einzelner Stunden von 1.1.2008 bis 31.12.2008

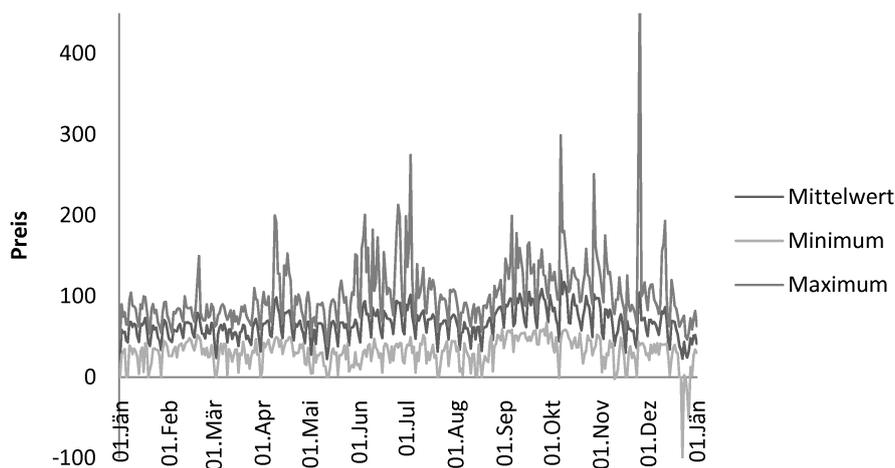


Abbildung 6: Zeitreihe der mittleren, minimalen und maximalen Tageswerte von 1.1.2008 bis 31.12.2008

2.2 Trendanalyse

Eine Trendanalyse dient dazu, Trends in Zeitreihen zu identifizieren. Diese lassen sich als langfristige Niveauveränderung der Zeitreihe beschreiben. Ein anschauliches Beispiel dafür ist die Weltbevölkerung, die kontinuierlich steigt. Dieser Schritt der Analyse ist wichtig um in weiterer Folge entscheiden zu können ob dies im Prognosemodell berücksichtigt werden soll. Die zwei gängigsten Methoden sind Trendregressionen sowie gleitende Mittelwerte. Die Trendregression liefert direkt ein quantitatives Ergebnis und ist daher für diese Zwecke am brauchbarsten. Folgende Regression wurde, um mögliche Trends in der Zeitreihe zu identifizieren aufgestellt. Der Schätzer wurde mit der Methode der kleinsten Quadrate berechnet.

$$y_t = \alpha + \beta_1 t + \beta_2 t^2 + \beta_3 t^3 \quad (2.1)$$

Aufgrund der großen Datenmenge konnte bei der gesamten Zeitreihe der kubische Trend mit dieser Methode nicht mehr berücksichtigt werden. Gerechnet wurde die Regression für die gesamte Zeitreihe, die Zeitreihen der einzelnen Stunden sowie der Zeitreihe der mittleren, maximalen und minimalen Preise. Das Ergebnis für ausgewählte Stunden befindet sich in Tabelle 2. Um die Ergebnisse der gesamten Zeitreihe mit jenen der einzelnen Stundenzeitreihen vergleichen zu können wurde statt t in der Regression $\frac{t}{24}$ verwendet. Alle geschätzten Parameter sind statistisch signifikant. Das Ergebnis zeigt für die Zeitreihen der einzelnen Stunden, dass der Wert des linearen Trend β_1 , des quadratischen Trend β_2 sowie des kubischen Trend β_3 über einen langen Zeitraum nicht vernachlässigbar wären. Für unsere Kurzzeitprognose kann man aber aufgrund der niedrigen Parameterwerte (die ja genau einen Tag entsprechen) einen Trend weglassen.

	α	t-Statistik	p-Wert	β_1	t-Statistik	p-Wert	β_2	t-Statistik	p-Wert	β_3	t-Statistik	p-Wert
03:00	28,7	12,5	1,62E-32	0,142	5,1	3,96E-07	-5,45E-04	-6,0	2,51E-09	4,51E-07	5,4	6,97E-08
10:00	51,2	12,7	1,15E-33	0,638	13,1	1,73E-35	-2,27E-03	-14,4	1,56E-41	1,98E-06	13,7	6,44E-38
20:00	49,9	28,6	1,29E-120	0,330	15,7	1,03E-47	-1,19E-03	-17,4	1,13E-56	1,00E-06	16,0	1,52E-49
Mittelwert	42,0	19,3	3,58E-67	0,375	14,3	7,54E-41	-1,34E-03	-15,8	2,24E-48	1,16E-06	14,8	1,38E-43
Gesamte Zeitreihe	63,3	110,0	0,00E+00	0,018	4,8	1,97E-06	-9,89E-05	-19,6	5,05E-85			

Tabelle 2: Ergebnisse der Trendregression

2.3 Autokorrelationsfunktion und partielle Autokorrelationsfunktion

Die Autokovarianzfunktion dient dazu, Zusammenhänge zwischen zwei Werten zu verschiedenen Zeitpunkten darzustellen und ist folgendermaßen definiert:

$$\gamma(r, s) = Cov(X_r, X_s)$$

X_r sowie X_s stellen die Prozesse zum Zeitpunkt r und s dar. Für stationäre Prozesse gilt $\gamma(r, s) = \gamma(r + t, s + t)$ und damit in weiterer Folge $\gamma(r - s) := \gamma(r - s, 0)$; Der Wert der Kovarianzfunktion ist also nicht von der Lage, sondern nur von Abstand zweier Werte abhängig. Daher gibt man bei stationären Prozessen immer nur den Abstand (Lag) zweier Werte als Parameter an. Die Autokorrelationsfunktion (ACF) mit Lag r ist definiert als:

$$\rho(r) = \frac{\gamma(r)}{\gamma(0)}$$

Die partielle Autokorrelationsfunktion (PACF) zeigt den Zusammenhang zwischen zwei Werten zu verschiedenen Zeitpunkten bereits bereinigt von den Werten dazwischen. Es bleibt also nur noch der Zusammenhang zwischen zwei Werten übrig und nicht wie bei der ACF auch noch der Zusammenhang mit den Werten dazwischen. Mit der gleichen Argumentation wie bei der ACF

lässt sich die PACF einer stationären Zeitreihe als Funktion von nur einem Parameter beschreiben. Dieser Parameter beschreibt den Lag zum Wert, der mit Null indiziert ist.

Der Partielle Autokorrelationskoeffizient mit Lag k ist $\alpha(k) = \phi_{kk}$ wobei ϕ_{kk} bestimmt ist durch:

$$\begin{bmatrix} \rho(0) & \rho(1) & \rho(2) & \dots & \rho(k-1) \\ \rho(1) & \rho(0) & \rho(1) & \dots & \rho(k-2) \\ \vdots & & & & \vdots \\ \rho(k-1) & \rho(k-2) & \rho(k-3) & \dots & \rho(0) \end{bmatrix} \begin{bmatrix} \phi_{k1} \\ \phi_{k2} \\ \vdots \\ \phi_{kk} \end{bmatrix} = \begin{bmatrix} \rho(1) \\ \rho(2) \\ \vdots \\ \rho(k) \end{bmatrix}$$

Mithilfe dieser Analyse kann man eventuell Rückschlüsse auf den Prozess ziehen, dessen Realisierung wir hier in den Spotpreisen sehen. Die ACF eines AR(p) Prozesses ist exponentiell abfallend oder gedämpft schwingend abfallend und die PACF bricht nach Lag p ab. Bei einem MA(q) Prozess ist es genau umgekehrt. ARMA Prozesse erkennt man dann durch das Vermischen der beiden Eigenschaften. Die beiden Funktionen wurden jeweils für die gesamte Zeitreihe sowie für die Zeitreihe der einzelnen Stunden gerechnet und graphisch dargestellt. Die Ergebnisse sieht man in den Abbildungen 7 bis 10 und lassen Rückschlüsse auf einen AR Prozess zu.

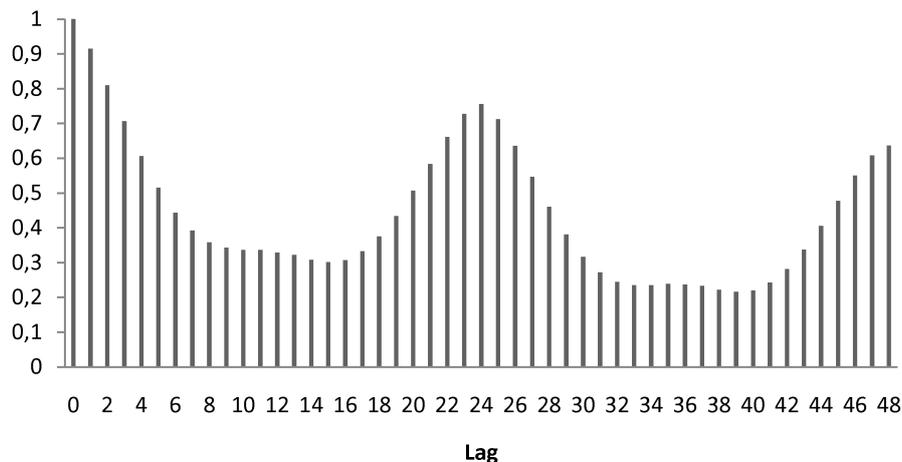


Abbildung 7: ACF der gesamten Zeitreihe

Lag 1 entspricht einer Stunde und man sieht deutlich, dass ein starker Zusammenhang zum Wert vor 24 Stunden also einen Tag besteht.

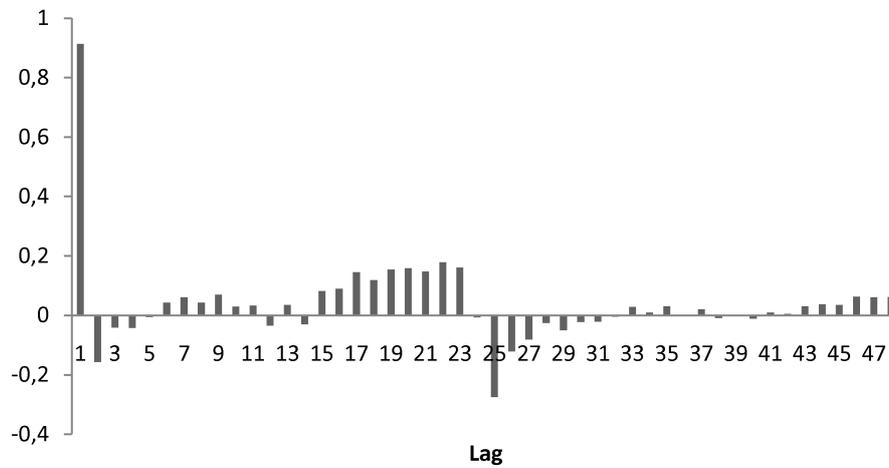


Abbildung 8: PACF der gesamten Zeitreihe

Lag 1 entspricht einer Stunde und leider lässt die Grafik keine eindeutigen Rückschlüsse zu.

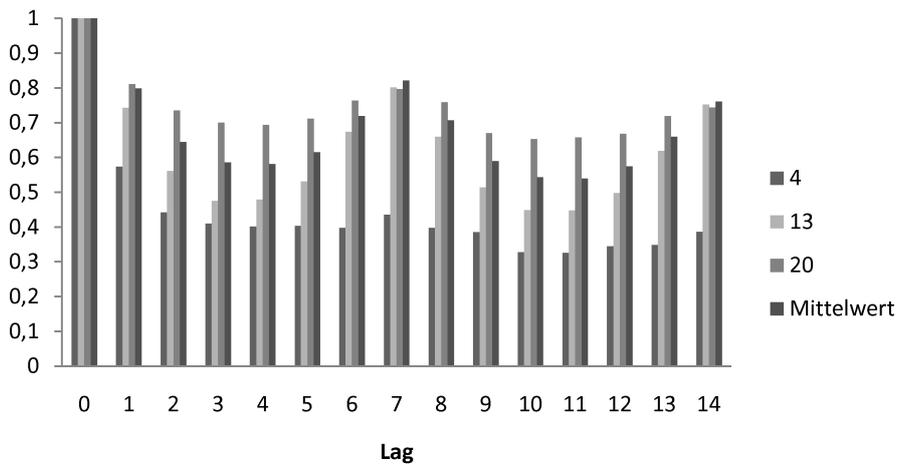


Abbildung 9: ACF einzelner Stunden

Lag 1 entspricht einem Tag und man sieht deutlich den Zusammenhang zum Wert von vor 7 Tagen also einer Woche.

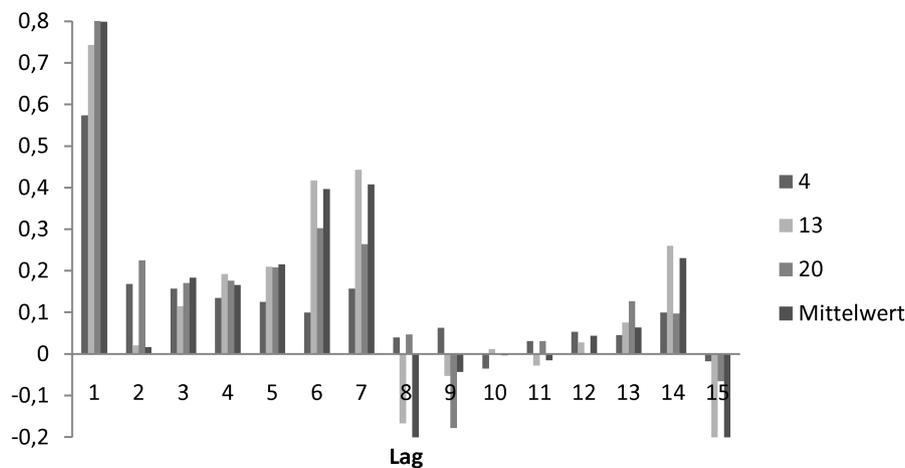


Abbildung 10: PACF einzelner Stunden

Lag 1 entspricht einem Tag und in dieser PACF erkennt man auch den Zusammenhang zum Wert von vor einer Woche.

2.4 Periodogramm

Periodizitäten kann man auf unterschiedliche Weise erkennen. Es gibt verschiedene Möglichkeiten mit Schwingungen beziehungsweise Dummies auf vermuteten Frequenzen zu testen. Eine weitere sehr weit verbreitete Methode ist das Periodogramm. Dies kann man verstehen als frequenzabhängige Kovarianzfunktion. Praktisch wird dies mithilfe einer Fouriertransformation der Kovarianzfunktion gemacht und als Ergebnis erhält man zu jeder vermuteten Frequenz die dazugehörige Intensität. Die entsprechende Formel, mit $\hat{\gamma}(s)$ aus der ACF ist:

$$I(\lambda) = T \left(\frac{1}{T} \sum_{t=1}^T (x_t - \bar{x}) e^{-i\lambda t} \right) \left(\frac{1}{T} \sum_{s=1}^T (x_s - \bar{x}) e^{i\lambda s} \right) = \sum_{s=-T+1}^{T-1} \hat{\gamma}(s) e^{-i\lambda s}$$

In der Zeitreihe der Spotpreise wurden unterschiedliche Frequenzen verwendet. Die gesamte Zeitreihe wurde untersucht und die Ergebnisse befinden sich in Abbildung 11. Ein starker Zusammenhang besteht also zur Schwingung mit Schwingungsdauer 12h und 24h. Als nächster Schritt wurden die Zeitreihen einzelner Stunden mittels Periodogramm untersucht. Der Fokus lag darauf die Zusammenhänge mit den Schwingungen mit Schwingungsdauer von sieben Tagen, zwei, vier und 20 Wochen sowie 365 Tagen zu untersuchen. Die genauen Ergebnisse sieht man in Abbildung 12. Als weiterer Analyseschritt wurde noch betrachtet, ob der Zusammenhang unterschiedlich stark in Abhängigkeit der Stunde ist. Die Ergebnisse sind durchaus unterschiedlich, wie man auch in Abbildung 13 sieht.

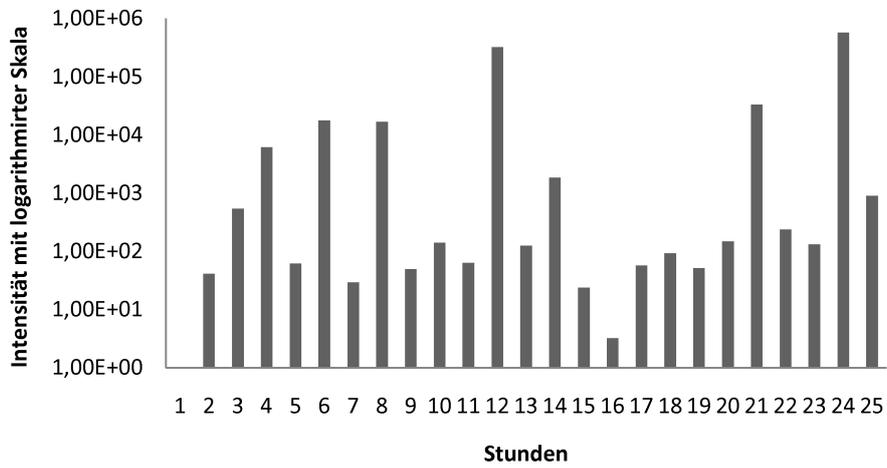


Abbildung 11: Periodogramm der gesamten Zeitreihe

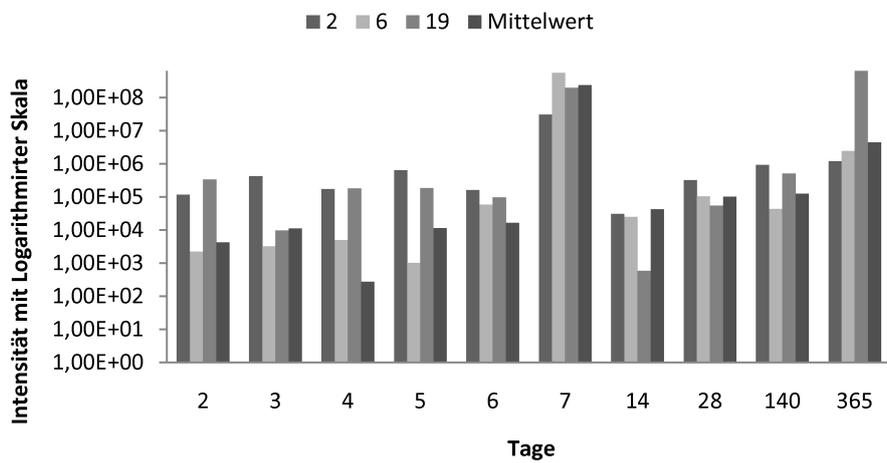


Abbildung 12: Periodogramm einzelner Stunden

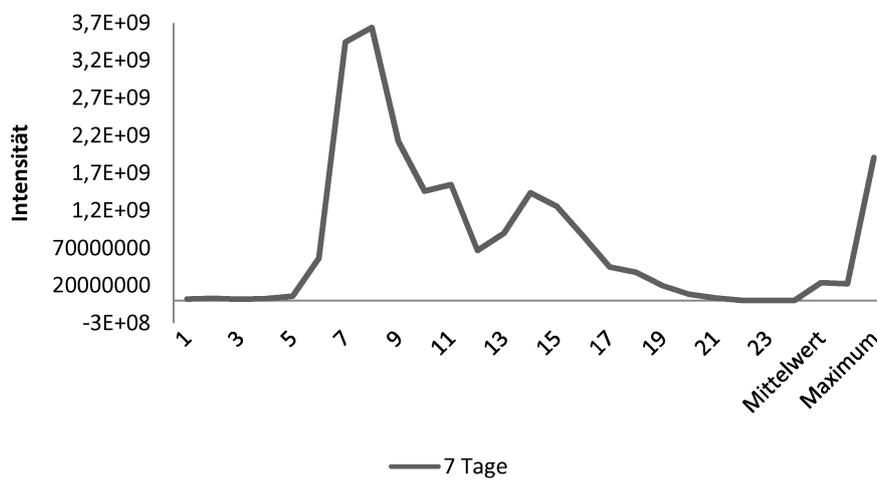


Abbildung 13: Intensität des Periodogramms in Abhängigkeit der gewählten Stunde

2.5 Gleitende Mittelwerte und Varianzen

Gleitende Mittelwerte und Varianzen werden verwendet um zu überprüfen, ob sich die jeweiligen Werte konstant über die Zeit sind, oder man weitere Annahmen treffen muss. Des weiteren lässt es eventuell Rückschlüsse zu, in wie weit Ausreißer Einfluss auf diese Werte nehmen. Als erster Schritt wurde analysiert, ob die Werte sich stundenweise ändern. Ist zum Beispiel der Mittelwert um 3h in Früh anders als um 14h am Nachmittags. Es stellte sich durch die Analyse heraus, dass die Mittelwerte und Varianzen, wie auch schon erwartet, sehr unterschiedlich in Bezug auf die jeweilige Stunde sind (siehe Abbildung 14 und 15, Global bezeichnet die gesamte Zeitreihe der jeweiligen Stunde). Im späteren Prognosemodell ist also auf jeden Fall ein Unterschied nach Stunden zu machen. Als nächster Schritt wurde untersucht, ob es Differenzen bei den Mittelwerten und Varianzen der einzelnen Tagestypen gibt. Als Tagestyp wurde vereinfacht angenommen, dass es nur drei verschiedene Typen gibt: Wochentag, Samstag und Sonntag bzw Feiertag (im weiteren werden für Sonn- und Feiertage einheitlich als Sonntag bezeichnet; Die Feiertage entsprechen den Bundesweiten Feiertagen Deutschlands). Das Ergebnis der Analyse war, dass die Mittelwerte der Wochentage höher sind als jene der Samstage, welche wieder höher sind als jene der Sonntage. Eine Ausnahme bilden hier die frühen Morgenstunden von ca 1h bis 5h. Hier sind die Mittelwerte der Sonntage höher. Dies erscheint auch durchaus plausibel, da zu dieser Uhrzeit am Wochenende auch mehr Menschen wach sind als unter der Woche und damit auch mehr Strom konsumieren (siehe Abbildung 14). Bei den Varianzen verhalten sich die Wochentage und Samstag ähnlich und liegen deutlich über den Werten von Sonntagen (siehe Abbildung 15). Weiters zu beachten ist aber, dass es einige deutliche Ausreißer gibt (z.B.: Wert Sonntags, Stunde 2) und es durchaus angebracht sein könnte einen robusteren Varianzschätzer zu verwenden. Es macht also durchaus Sinn im späteren Prognosemodell eine Unterscheidung nach Wochentagen einzubauen sowie einen verallgemeinerten kleinste Quadrate Schätzer zu verwenden um auch Tagesunterschiedliche Varianzen einzubauen. Dies bedeutet aber auch, dass ein Modell mit fixen Lag-Parametern wie zum Beispiel AR Modelle nicht ausreichend für eine gute Prognose sein können.

Als letzter Schritt wurden über die knapp zwei Jahre lange Zeitreihe 15 Fenster gelegt und untersucht, ob sich Varianzen und Mittelwerte im Laufe der Zeit verändern. Die Werte schwanken relativ stark und es zeigt erneut, dass im späteren Modell angedacht werden sollte, die Varianzen abhängig vom Zeitpunkt zu machen (siehe Abbildungen 16 und 17). Des weiteren gibt es wieder einige Ausreißer.

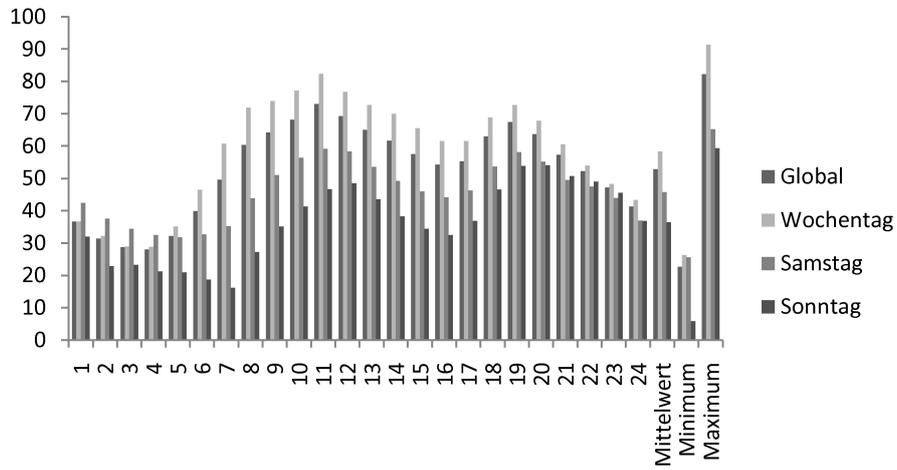


Abbildung 14: Mittelwerte(y-Achse) der einzelnen Stunden(x-Achse) abhängig vom Tagestyp

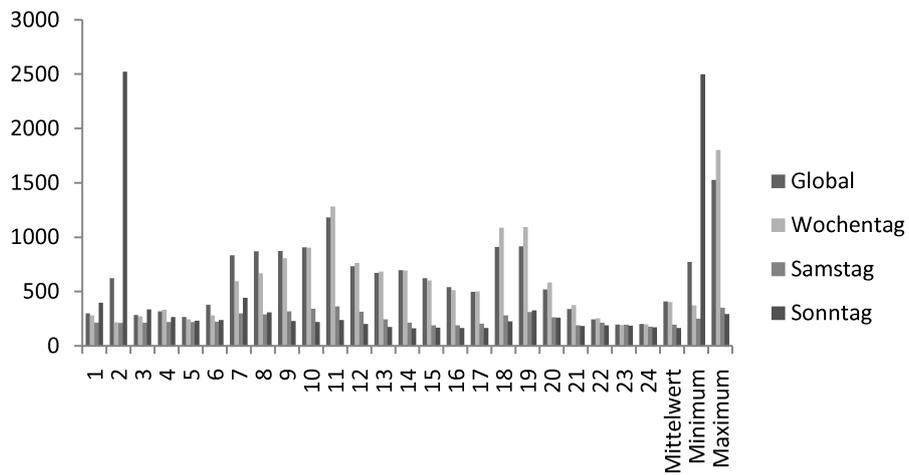


Abbildung 15: Varianzen(y-Achse) der einzelnen Stunden(x-Achse) abhängig vom Tagestyp

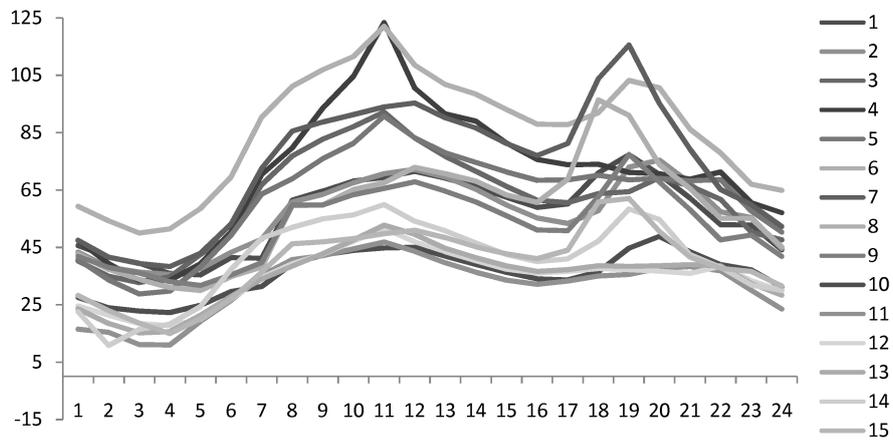


Abbildung 16: Mittelwerte abhängig von der Stunde mit 15 gleitenden Fenstern

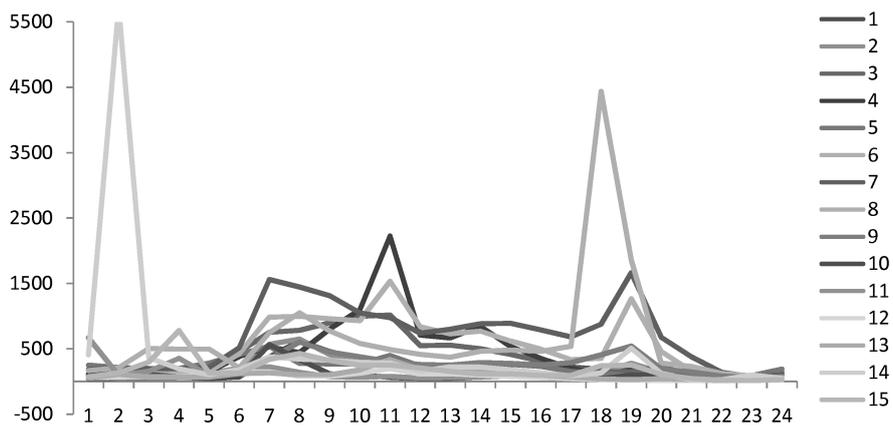


Abbildung 17: Varianzen abhängig von der Stunde mit 15 gleitenden Fenstern

3 Exogene Variablen

Zusätzlich zu den Preisdaten sollen in die Prognosemodelle auch noch exogene Variablen einfließen. Die Fragestellung dabei ist in wie weit externe Variablen einen Einfluss auf den Strompreis des nächsten Tages haben. In dieser Arbeit wurden vier Variablen ausgewählt die in den späteren Modellen berücksichtigt werden sollen.

3.1 Wind

Die Windkraftprognosen werden immer wichtiger für eine gute Prognose der Stromspotpreise. Das liegt einerseits daran, dass es immer mehr Windturbinen gibt und andererseits daran, dass die Leistung sehr stark abhängig vom Wind ist. Die erbrachte Leistung steigt exponentiell mit dem vorhandenen Wind und bricht dann ab einen gewissen Wert auf Null ein, da man die Windkraftwerke ausschalten muss. Die Genauigkeit der Vorhersage ist also besonders wichtig. e&t verwendet als Quelle für die Windkraftprognosen Eurowind [Wind], die mehrmals täglich Windprognosen für Deutschland veröffentlichen. Die Datengrundlage für diese Arbeit ist eine von e&t Research teilweise modifizierte Prognosezeitreihe die wie die Strompreise stündlich verfügbar ist. In Abbildung 18 befindet sich der Ausschnitt von einer Windkraftprognose in MW für den Zeitraum von einer Woche mit logarithmierter Skala.

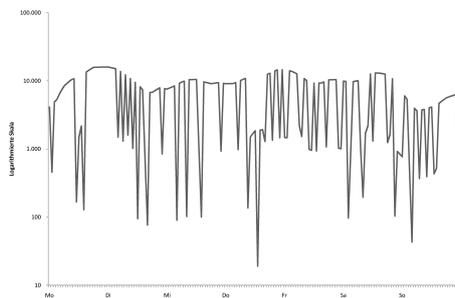


Abbildung 18: Windprognose vom 7.1.2008 bis 13.1.2008

3.2 Temperatur

Die Temperaturprognose beinhaltet die erwarteten Mittleren Temperaturen für den nächsten Tag in Deutschland und die Werte stammen von e&t Research

3.3 Lastprognosen

Mithilfe von e&t Research wurde aus verschiedenen Quellen ein Datensatz für die täglichen Lastprognosen erstellt. Die Prognose beinhaltet die erwarteten eingesetzten Kraftwerke aufgeteilt auf Kernenergie, Braunkohle, Steinkohle, Gas, Wasser und Sonstige Kraftwerke in MW. Für das Prognosemodell wurde die Summe gebildet und als Variable Einsatz definiert. Besonders interessant ist es auch diese Variable als Näherungswert des Stromangebots am Markt

zu sehen. Des Weiteren stehen noch die gebauten, funktionstüchtigen Kraftwerke (was aber nicht unbedingt heißt, dass diese auch in Betrieb sind) zur Verfügung; diese Variable bringt aber relativ wenig da sie keine Rückschlüsse auf Angebot zulässt und auch weiters keine zusätzlichen für die Prognose relevanten Informationen enthält. Als dritter Wert wurden noch die tatsächlich verwendeten Kraftwerke angegeben. Dieser Wert ist aber für eine Prognose nicht sehr sinnvoll, da er immer erst im Nachhinein verfügbar ist und das Hauptwerk dieser Arbeit doch darauf liegt ein gutes Prognosemodell zu konzipieren.

3.4 Strombörse EXAA

Alle Informationen dieses Kapitels stammen aus verschiedenen Dokumenten von der Energy Exchange Austria (EXAA) Homepage sowie aus Texten von der Homepage. Genauere Angaben dazu befinden sich im Literaturverzeichnis unter der Referenz [EXAA2]. Die EXAA gibt es seit 2001. 2002 wurde der Spothandel mit Strom aufgenommen. Der Handel erfolgt ähnlich wie an der EEX, jedoch wird aufgrund der Größe der EXAA und Rücksichtnahme auf kleinere Unternehmen nur an Werktagen gehandelt. Am Freitag wird also zum Beispiel der Strom für den regulären Folgetag Samstag, sowie Sonntag und Montag gehandelt. Der Vorteil der EXAA besteht darin, dass bereits um 10:15 die Preise veröffentlicht werden und sie somit als sehr guter Indikator für alle weiteren Auktionen und Handel gilt. Marktgebiet für die Stromauktionen sind alle Regelzonen Deutschlands und Österreichs. Das Volumen, das 2009 gehandelt wurde entspricht 4,66TWh und ist somit um einiges kleiner als jenes der EEX, aber dennoch stabil genug um als gute Preisreferenz zu dienen.

4 Prognose: Einleitung und theoretische Grundlagen

4.1 Prognosemodelle

In den folgenden vier Abschnitten werden verschiedene Modelle vorgestellt, die für die spätere Prognose verwendet werden. Ziel dabei war es einerseits, die Prognose mit den reinen Preisdaten von jener mit exogenen Variablen zu trennen und andererseits für die kleinste Quadratschätzung ein Modell ohne und eines mit Kreuztermen zu bauen.

4.1.1 Einfaches lineares Modell ohne exogene Variablen

Im Kapitel der Zeitreihenanalyse wurden schon einige Ideen für die Prognosemodelle angeregt. Eine Unterscheidung nach Tagestyp erscheint dabei unbedingt notwendig. Dadurch stößt man bei einem klassischen AR Modell aber schon auf ein Problem. Nimmt man zum Beispiel die Lags 1 bis 7 in die Analyse, so ist am Montag der Lag 1 ein Sonntag und am Dienstag der Lag 1 ein Montag. Dadurch ist es aufgrund der Datenstruktur schon unmöglich die Koeffizienten zufriedenstellend zu schätzen. Eine einfache und sehr wirksame Modifikation besteht darin Tagestypabhängig die Lags einzubauen. An einem Werktag wird also zum Beispiel immer der Wert des vorherigen Werkta-

ges verwendet. Dazu werden drei Tagestypen definiert: Werktag, Samstag und Sonntag, wobei Feiertage wie Sonntage behandelt werden. Weiters wurde oben schon die Wochenstruktur besprochen; der Wert des letzten Tages des gleichen Types hat einen großen Einfluss auf den heutigen Wert, sowie auch der Wert der vorangegangenen Woche und des letzten Jahres. Nachdem für die Analyse die Daten von knapp zwei Jahren zur Verfügung stehen macht eine Einbeziehung des Letzjahreswert wenig Sinn. Insgesamt kommt man also zur Auswahl der folgenden Parameter, den Wert des letzten Tages gleichen Types, sowie den Durchschnitt der letzten fünf Tage gleichen Types und den der letzten sieben Tage zu betrachten. Dieser wurde gewählt, um zu überprüfen ob eventuell durch das allgemeine Niveau der später erklärten Dummies die Tagesstruktur schon ausreichend abgebildet ist und ein zusätzliches Niveau der letzten sieben Wochentage ausreicht oder ob eine genauere Unterscheidung notwendig ist. Alternativ könnte man auch den Wert des Vortages nehmen. Dies führt aber zu Verzerrungen da der Wert sich an einigen Tagen natürlich mit dem Wert des Vortages gleichen Types deckt und keine eindeutige Zuordnung der Phänomäne erfolgt. Um allgemein noch ein tagesspezifisches Niveau einzuführen werden Dummies verwendet: Das Allgemeine Niveau beziehungsweise Intercept sowie ein Dummy für Samstag und Sonntag und einer für Sonntag werden verwendet. Die Dummies wurden so gewählt, um eventuell einen Rückschluss darauf zu bekommen, ob eine Unterscheidung zwischen Samstag und Sonntag überhaupt notwendig ist. Zusammenfassend werden also folgende Variablen eingeführt:

Koeffizient	erklärende Variable	Kurzbeschreibung
θ_0	1	allgemeines Niveau
θ_1	δ_{SaSo}	Dumme für zusätzliches Niveau am Samstag und Sonntag
θ_2	δ_{So}	Dumme für zusätzliches Niveau am Sonntag
θ_3	\bar{Y}_T	Durchschnitt der letzten 5 Tage gleichen Types
θ_4	Y_T	Wert des letzten Tages gleichen Types
θ_5	\bar{Y}_7	Durchschnitt der letzten 7 Tage

Tabelle 3: Erklärende Variablen der Preise

Ein erstes einfaches lineares Prognose Modell lautet also:

$$Y(t) = \theta_0 + \theta_1 \delta_{SaSo}(t) + \theta_2 \delta_{So}(t) + \theta_3 \bar{Y}_T(t) + \theta_4 Y_T(t) + \theta_5 \bar{Y}_7(t) + u(t) \quad (4.1)$$

$u(t)$ beschreibt dabei den Fehlerterm zum Zeitpunkt t .

4.1.2 Einfaches lineares Modell mit exogenen Variablen

Als weiterer Schritt wurde von der e&t angeregt exogene Variablen in die Prognose einzubauen. Diese wurden schon kurz in Kapitel 3 erklärt. Für die Prognose standen vier verschiedene Werte zur Verfügung: Die Prognose für den Wind, die Temperaturprognose, die Prognose für die eingesetzten Kraftwerke

und die Werte der Strombörse EXAA, welche alle nur an Werktagen aktualisiert werden, aber für alle sieben Tage verfügbar sind. Die Werte der EXAA sind auf stündlicher Basis verfügbar, jene für Wind, Einsatzprognose sowie Temperatur nur für den ganzen Tag. Zusammenfassend hat man also folgende zusätzliche Variablen zur Verfügung:

Koeffizient	erklärende Variable	Kurzbeschreibung
θ_6	Wind	Windprognose
θ_7	Einsatz	Prognose der verfügbaren Kraftwerkskapazitäten
θ_8	Temp	Temperaturprognose
θ_9	EXAA	Wert an der Strombörse EXAA für denselben Tag

Tabelle 4: Exogene Variablen

Dies führt zu einem erweiterten linearen Prognosemodell:

$$Y(t) = \theta_0 + \theta_1 \delta_{SaSo}(t) + \theta_2 \delta_{So}(t) + \theta_3 \bar{Y}_T(t) + \theta_4 Y_T(t) + \theta_5 \bar{Y}_7(t) + \theta_6 \text{Wind}(t) + \theta_7 \text{Einsatz}(t) + \theta_8 \text{Temp}(t) + \theta_9 \text{EXAA}(t) + u(t) \quad (4.2)$$

4.1.3 Großes Modell ohne exogene Variablen

Eine weitere Überlegung ist es, Kreuzterme einzubauen, um die Qualität der Regression zu verbessern. Dazu werden für alle Variablen die in Abschnitt "Einfaches Lineares Modell ohne exogene Variablen" beschrieben wurden die Kreuzprodukte mit sich selbst und untereinander gebildet. Besondere Vorsicht ist hier bei den Dummi-Variablen angebracht. Quadrieren von Dummi-Variablen und Multiplizieren untereinander macht keinen Sinn und führt außerdem zu einem nicht lösbares Regressionsproblem. Beim Konzipieren des Modells muss dies also berücksichtigt werden. Insgesamt erhält man 17 Variablen plus Intercept. Das Modell lautet somit:

$$Y(t) = \theta_0 + \theta_1 \delta_{SaSo}(t) + \theta_6 \delta_{SaSo}(t) \bar{Y}_T(t) + \theta_7 \delta_{SaSo}(t) Y_T(t) + \theta_8 \delta_{SaSo}(t) \bar{Y}_7(t) + \theta_2 \delta_{So}(t) + \theta_9 \delta_{So}(t) \bar{Y}_T(t) + \theta_{10} \delta_{So}(t) Y_T(t) + \theta_{11} \delta_{So}(t) \bar{Y}_7(t) + \theta_3 \bar{Y}_T(t) + \theta_{12} \bar{Y}_T(t) \bar{Y}_T(t) + \theta_{13} \bar{Y}_T(t) Y_T(t) + \theta_{14} \bar{Y}_T(t) \bar{Y}_7(t) + \theta_4 Y_T(t) + \theta_{15} Y_T(t) Y_T(t) + \theta_{16} Y_T(t) \bar{Y}_7(t) + \theta_5 \bar{Y}_7(t) + \theta_{17} \bar{Y}_7(t) \bar{Y}_7(t) + u(t) \quad (4.3)$$

Des weiteren könnte man noch andedenken, die quadrierten Terme mit Tages-spezifischen Dummies zu versehen. Dies wurde in dieser Arbeit aber nicht betrachtet.

4.1.4 Großes Modell mit exogenen Variablen

Denselben Schritt wie im vorangegangenen Unterabschnitt kann man auch mit dem Modell aus dem Abschnitt "Einfaches Lineares Modell mit exogenen Variablen" machen und kommt dann auf ein Modell mit insgesamt 51 sinnvollen Variablen:

$$\begin{aligned}
Y(t) = & \\
& \theta_0 + \theta_1 \delta_{SaSo}(t) + \theta_{10} \delta_{SaSo}(t) \bar{Y}_T(t) + \theta_{11} \delta_{SaSo}(t) Y_T(t) + \theta_{12} \delta_{SaSo}(t) \bar{Y}_7(t) \\
& + \theta_{13} \delta_{SaSo}(t) Wind(t) + \theta_{14} \delta_{SaSo}(t) Einsatz(t) + \theta_{15} \delta_{SaSo}(t) Temp(t) + \theta_{16} \delta_{SaSo}(t) EXAA(t) \\
& + \theta_2 \delta_{So}(t) + \theta_{17} \delta_{So}(t) \bar{Y}_T(t) + \theta_{18} \delta_{So}(t) Y_T(t) + \theta_{19} \delta_{So}(t) \bar{Y}_7(t) \\
& + \theta_{20} \delta_{So}(t) Wind(t) + \theta_{21} \delta_{So}(t) Einsatz(t) + \theta_{22} \delta_{So}(t) Temp(t) + \theta_{23} \delta_{So}(t) EXAA(t) \\
& + \theta_3 \bar{Y}_T(t) + \theta_{24} \bar{Y}_T(t) \bar{Y}_T(t) + \theta_{25} \bar{Y}_T(t) Y_T(t) + \theta_{26} \bar{Y}_T(t) \bar{Y}_7(t) \\
& + \theta_{27} \bar{Y}_T(t) Wind(t) + \theta_{28} \bar{Y}_T(t) Einsatz(t) + \theta_{29} \bar{Y}_T(t) Temp(t) + \theta_{30} \bar{Y}_T(t) EXAA(t) \\
& + \theta_4 Y_T(t) + \theta_{31} Y_T(t) Y_T(t) + \theta_{32} Y_T(t) \bar{Y}_7(t) + \theta_{33} Y_T(t) Wind(t) \\
& + \theta_{34} Y_T(t) Einsatz(t) + \theta_{35} Y_T(t) Temp(t) + \theta_{36} Y_T(t) EXAA(t) \\
& + \theta_5 \bar{Y}_7(t) + \theta_{37} \bar{Y}_7(t) \bar{Y}_7(t) + \theta_{38} \bar{Y}_7(t) Wind(t) + \theta_{39} \bar{Y}_7(t) Einsatz(t) \\
& + \theta_{40} \bar{Y}_7(t) Temp(t) + \theta_{41} \bar{Y}_7(t) EXAA(t) \\
& + \theta_6 Wind(t) + \theta_{42} Wind(t) Wind(t) + \theta_{43} Wind(t) Einsatz(t) \\
& + \theta_{44} Wind(t) Temp(t) + \theta_{45} Wind(t) EXAA(t) \\
& + \theta_7 Einsatz(t) + \theta_{46} Einsatz(t) Einsatz(t) + \theta_{47} Einsatz(t) Temp(t) + \theta_{48} Einsatz(t) EXAA(t) \\
& + \theta_8 Temp(t) + \theta_{49} Temp(t) Temp(t) + \theta_{50} Temp(t) EXAA(t) \\
& + \theta_9 EXAA(t) + \theta_{51} EXAA(t) EXAA(t) + u(t)
\end{aligned}$$

4.2 Prognoseverfahren

4.2.1 Naive Prognose

Bei der naiven Prognose wird der Wert des Vortages als Prognosewert herangezogen. Dies ist eine recht einfache Methode, die bei Prognosen oft gute Resultate bringt und bereits einen Wert liefert der schwer zu schlagen ist. In dieser Arbeit wird die naive Prognose als Benchmark definiert und Ziel ist es, bessere Prognosen als die Benchmark zu machen. Weiters wird die Wochenstruktur eingebaut und die naive Prognose definiert als Wert des Vortages gleichen Types. Damit wird es nochmal schwieriger die Benchmark zu schlagen. In Abbildung 19 sieht man, wie die naive Prognose für einen Zeitraum von vier Wochen aussieht.

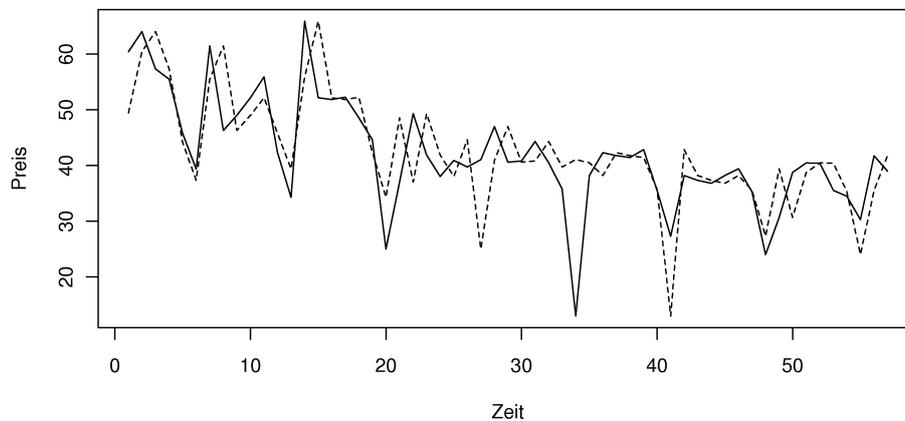


Abbildung 19: Naive Prognose

4.2.2 Die Methode der kleinsten Quadrate

Die kleinste Quadrate Methode ist in linearen Modellen eine gängige Methode zum Schätzen der Parameter. Es werden dabei jene Parameter gewählt, welche die Residuenquadratsumme minimieren. Ein lineares Modell hat beispielsweise folgende Form:

$$y = X\beta + u$$

y ist der Vektor zu erklärenden Variable, X die Matrix mit den erklärenden Variablen, β der Parametervektor und u dem Fehlervektor. Es gilt die Residuenquadratsumme zu minimieren:

$$\min u^2 = \min u'u = \min (y - X\beta)'(y - X\beta)$$

Durch Ableiten nach dem Parametervektor und anschließendem Nullsetzen erhält man unter der Annahme, dass X vollen Rang hat:

$$\hat{\beta} = (X'X)^{-1}X'y$$

4.2.3 Die Methode der kleinsten Quadrate mit Gewichten

Die Preise zeigen eine starke tagesabhängige Struktur und auch tagesspezifische Varianzen. Eine mögliche Berücksichtigung der tagesspezifischen Varianzen im Modell liefert das verallgemeinerte Regressionsmodell mit heteroskedastischen Fehlern. Dabei wird die Varianz nicht mehr als konstant über alle Beobachtungen angesehen, sondern eine Diagonalmatrix angegeben die zu jedem Zeitpunkt die zugehörige Varianz angibt. Es wird also eine Varianz geschätzt, die abhängig vom Tagestyp ist und in der Zukunft daher eindeutig für die Rücktransformation der Prognosewerte zuordenbar ist. Damit erhält man eine Diagonalmatrix, die abhängig vom Tagestyp eine andere Varianz enthält. Zusammenfassend ist die prinzipielle Idee Beobachtungen entsprechend eines σ abhängig vom Tagestyp zu gewichten.

4.2.4 Support Vector Regression

Die Support Vector Regression basiert auf der Idee von Support Vector Machines, welche in "A Tutorial on Support Vector Regression" [SVR1] ausführlich erklärt wird. Für die geschichtliche Entwicklung verweise ich ebenfalls auf dieses Paper. Die Grundidee stammt von V. Vapnik [SVR2] und lautet folgendermaßen:

Wir nehmen an, dass X unsere erklärenden Variablen (T Beobachtungen und k erklärende Variablen) enthält und y der zu erklärende Vektor ist. Unser Ziel in der ϵ -Support Vector Regression ist es, eine Funktion $f(x)$ zu finden, die maximal ϵ Abweichung von unserem Zielvektor y hat. Für den linearen Fall nimmt das Problem folgende Form an:

$$f(x) = \langle w, x \rangle + b \text{ mit } w \in \mathbb{R}^k, b \in \mathbb{R}$$

$\langle \cdot, \cdot \rangle$ ist dabei das Skalarprodukt in \mathbb{R}^k . Weiters möchte man eine möglichst flache Funktion finden, also jene die $\langle w, w \rangle$ minimiert. Insgesamt erhält man folgendes konvexes Optimierungsproblem:

$$\begin{aligned} \text{Minimieren von } (w) \quad & \frac{1}{2} \|w\|^2 \\ \text{unter} \quad & y_i - \langle w, x_i \rangle - b \leq \epsilon \\ & \langle w, x_i \rangle + b - y_i \leq \epsilon \end{aligned}$$

Das Problem bei obiger Formulierung liegt darin, dass eine derartige Funktion nicht existieren muss. Ein möglicher Ausweg dafür ist es einen Strafterm einzuführen, der Werte außerhalb des ϵ -Schlauch bestraft. Mit einem linearen Strafterm und den dazugehörigen Schlupfvariablen (ζ_i, ζ_i^*) für das Minimierungsproblem wird die Aufgabe zu:

$$\begin{aligned} \text{Minimieren von } (w, \zeta_i, \zeta_i^*): \quad & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^T (\zeta_i + \zeta_i^*) \\ \text{unter} \quad & y_i - \langle w, x_i \rangle - b \leq \epsilon + \zeta_i \\ & \langle w, x_i \rangle + b - y_i \leq \epsilon + \zeta_i^* \\ & \zeta_i, \zeta_i^* \geq 0 \end{aligned} \quad (4.4)$$

Die Konstante $C > 0$ beschreibt die Bereitschaft dafür, Abweichungen zu tolerieren. Die Ausreißer werden durch diesen Ansatz in der Modellierung besser als in der kleinsten Quadrate Regression berücksichtigt, da Werte die abweichen linear und nicht quadratisch bestraft werden (Ausreißer fließen also weniger stark in Prognose ein). Daher erscheint die Support Vector Regression für die Prognose von Strompreisen auch als eine geeignete Methode. Die Lösung dieses Problems geht oft einfacher über die duale Formulierung mit dem Lagrange Ansatz. Durch Einführen der positiven Lagrangemultiplikatoren erhält man:

$$\begin{aligned}
L := & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^T (\zeta_i + \zeta_i^*) - \sum_{i=1}^T (\eta_i \zeta_i - \eta_i^* \zeta_i^*) \\
& - \sum_{i=1}^T \alpha_i (\epsilon + \zeta_i - y_i + \langle w, x_i \rangle + b) \\
& - \sum_{i=1}^T \alpha_i^* (\epsilon + \zeta_i^* + y_i - \langle w, x_i \rangle - b) \tag{4.5}
\end{aligned}$$

Durch Ableiten, Nullsetzen und wieder Einsetzen in 4.4 kommt man zu der dualen Formulierung des Problems 4.4, die folgendermaßen aussieht:

$$\begin{aligned}
\text{Maximieren von } (\alpha_i, \alpha_i^*): & \quad -\frac{1}{2} \sum_{i,j=1}^T (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle x_i, x_j \rangle \\
& \quad -\epsilon \sum_{i=1}^T (\alpha_i - \alpha_i^*) - \sum_{i=1}^T y_i (\alpha_i - \alpha_i^*) \\
\text{unter} & \quad \sum_{i=1}^T (\alpha_i - \alpha_i^*) = 0 \text{ und } \alpha_i, \alpha_i^* \in [0, C] \tag{4.6}
\end{aligned}$$

Der Vorteil dieser Formulierung liegt darin, dass w nicht mehr explizit ermittelt werden muss, sondern nur noch das Skalarprodukt zwischen zwei Termen bekannt sein muss. Diese Transformation macht es in weiterer Folge einfacher auch nichtlineare Funktionen einzubauen. Dazu ersetzt man das Skalarprodukt $\langle x_i, x_j \rangle$ durch $k(x_i, x_j) := \langle \Phi(x_i), \Phi(x_j) \rangle$. Im weiteren wird $k(x_i, x_j)$ auch als Kernel bezeichnet. Einfacher ausgedrückt wird $f(x) = \langle w, x \rangle$ zu $f(x) = \langle w, \phi(x) \rangle$ also die Variablen werden z.B. nichtlinear transformiert, das Skalarprodukt an sich aber nicht beeinflusst. Besonderer Vorteil dabei ist es, dass man auch Φ nicht explizit kennen muss, sondern nur das Skalarprodukt zwischen zwei Elementen. Insgesamt erhält man:

$$\begin{aligned}
\text{Maximieren von } (\alpha_i, \alpha_i^*): & \quad -\frac{1}{2} \sum_{i,j=1}^T (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) k(x_i, x_j) \\
& \quad -\epsilon \sum_{i=1}^T (\alpha_i - \alpha_i^*) - \sum_{i=1}^T y_i (\alpha_i - \alpha_i^*) \\
\text{unter} & \quad \sum_{i=1}^T (\alpha_i - \alpha_i^*) = 0 \text{ und } \alpha_i, \alpha_i^* \in [0, C] \tag{4.7}
\end{aligned}$$

Durch Einsetzen in die First Order Condition, resultierend aus der Lagrangefunktion aus Gleichung 4.5, erhält man $f(x)$ sowie w

$$\begin{aligned}
w &= \sum_{i=1}^T (\alpha_i - \alpha_i^*) \Phi(x_i) \\
f(x) &= \sum_{i=1}^T (\alpha_i - \alpha_i^*) k(x_i, x) + b
\end{aligned} \tag{4.8}$$

Sind α_i und α_i^* gleich Null so liegt x_i innerhalb des ϵ -Bereichs. Jene x_i , für die zumindest α_i oder α_i^* ungleich Null sind, werden Support Vectors genannt. Es gibt also umso mehr Support Vectors, je enger der ϵ -Bereich ist.

Eine weitere Formulierung der Support Vector Regression wird in "Support Vector Machines in R" [SVR3] genannt. Diese zeichnet sich dadurch aus, dass C durch $\frac{C}{\nu\epsilon}$ ersetzt wird, daher zusätzlich über ϵ minimiert wird und daher bei der dualen Formulierung ϵ in die Nebenbedingung wandert und nicht mehr explizit angegeben werden muss. Die neue Formulierung des Support Vector Regression Problems lautet:

$$\begin{aligned}
\text{Minimieren von } (w, \epsilon, \zeta_i, \zeta_i^*): & \quad \frac{1}{2} \|w\|^2 + \frac{C}{\nu\epsilon} \frac{1}{T} \sum_{i=1}^T (\zeta_i + \zeta_i^*) \\
\text{unter} & \quad \begin{aligned}
\langle \Phi(x_i), w \rangle + b - y_i & \geq \epsilon - \zeta_i \\
y_i - (\langle \Phi(x_i), w \rangle + b) & \leq \epsilon - \zeta_i^* \\
\zeta_i, \zeta_i^* \geq 0, \epsilon & \geq 0
\end{aligned}
\end{aligned} \tag{4.9}$$

und deren duale Formulierung:

$$\begin{aligned}
\text{Maximieren von } (\alpha_i^*, \alpha_i) : & \quad \sum_{i=1}^T y_i (\alpha_i^* - \alpha_i) - \frac{1}{2} \sum_{i,j=1}^T (\alpha_i^* - \alpha_i) (\alpha_j^* - \alpha_j) k(x_i, x_j) \\
\text{unter} & \quad \sum_{i=1}^T (\alpha_i - \alpha_i^*) = 0 \text{ und } \alpha_i, \alpha_i^* \in \left[0, \frac{C}{k}\right] \\
& \quad \sum_{i=1}^T (\alpha_i + \alpha_i^*) \leq C\nu
\end{aligned} \tag{4.10}$$

In [SVR1] wird erklärt, welche Kernels zulässig für die Support Vector Regression sind. Im späteren Abschnitt über die verwendeten Bibliotheken in R werden die verwendeten Kernels kurz angesprochen.

4.2.5 Support Vector Regression mit Gewichten

Die Grundidee ist diesselbe wie der kleinsten Quadrate Schätzung. Das Problem ist aber noch komplexer, da man nicht nur eine einfache Regression rechnet sondern durch die Kernels noch ganz andere Funktionen bekommt. Trotzdem wurde nach einigen Überlegungen entschieden, dass derselbe Ansatz wie beim Verallgemeinerten Regressionsmodell verwendet wird, ungeachtet der Struktur der Kernels. Die Variablen werden also mit einer tagesspezifischen Varianz skaliert, die Prognose wie bei der normalen Support Vector Regression gerechnet und die Prognostizierten Werte rücktransformiert.

4.3 Qualitätskriterien

Die verschiedenen Qualitätskriterien dienen dazu, ein Maß zu bekommen, in wie weit die Prognose von den tatsächlichen Werten abweicht. Als Benchmark, die es zu schlagen gibt, wurde immer die naive Prognose genommen, welche schon weiter oben besprochen wurde. Die Prognose gilt somit als besser wenn sie in allen drei in Folge diskutierten Qualitätskriterien die Benchmark schlägt.

4.3.1 Mittlere Absolute Abweichung

Die mittlere absolute Abweichung der Prognose ($\hat{y}(t)$) vom tatsächlichen Wert ($y(t)$) ist definiert als:

$$MAD = \frac{1}{T} \sum_{t=1}^T |y(t) - \hat{y}(t)|$$

Vorteil im Gegensatz zur quadratischen Abweichung ist, dass die MAD in Bezug auf Ausreißer weniger sensibel ist.

4.3.2 Mittlere quadratische Abweichung

Die mittlere quadratische Abweichung der Prognose ($\hat{y}(t)$) vom tatsächlichen Wert ($y(t)$) ist definiert als:

$$MSD = \frac{1}{T} \sum_{t=1}^T (y(t) - \hat{y}(t))^2$$

4.3.3 Hitrate

Eine weitere Variante den Fehler bezüglich des tatsächlichen Wertes zu quantifizieren ist die Hitrate. Diese definiert sich dadurch, dass ausgehend vom heutigen Wert geschaut wird ob die Prognose einen höheren oder niedrigeren Wert angibt. Stimmt diese Richtung mit jener des tatsächlichen Wertes überein, so erhöht sich der Wert um eins. Das Endergebnis ist dann ein Prozentsatz der erfolgreichen Prognosen. Als Formel heißt das konkret:

$$HIT = \frac{1}{T} \sum_{t=2}^T I((y(t) - \hat{y}(t-1))((y(t) - y(t-1)) > 0)$$

Dabei bezeichnet I die Indikatorfunktion welche $I = 1$ für eine wahre Aussage ist und $I = 0$ für eine falsche Aussage ist.

Als Benchmark werden z.B. bei Aktienkursen immer 50% definiert. Eine Hitrate von über 50% ist also ein Erfolg was bei Strompreisen aber weniger Sinn macht. Ändert sich der Tagestyp, dann weiß man sehr oft schon die Richtung des Folgetages. Als Benchmark wird deswegen die Hitrate der naiven Prognose verwendet, welche die Tagesstruktur bereits berücksichtigt.

4.3.4 Autokorrelation der Fehler

Als letzter Schritt wird überüft, ob das Modell bei der kleinsten Quadrate Schätzung sowie das der verallgemeinerten kleinsten Quadrate Schätzung die Datenstruktur richtig erfasst. Dazu wird die Autokorrelation der Prognosefehler angeschaut, ist diese zwischen $\frac{\pm 2}{\sqrt{T}}$ so kann mit 95 % Wahrscheinlichkeit gesagt werden, dass die Autokorrelation Null ist.

4.4 AIC Schleife

Ein Problem bei großen Modellen ist, dass diese durch die vielen Variablen das Problem überdefinieren (die Effizienz der Schätzer wird schlechter) und eine Variablenselektion sinnvoll wird. Dazu wurde in dieser Arbeit das Akaike Informationskriterium verwendet. Dieses versucht die Anzahl der gewählten Parameter mit der Anpassungsgüte abzuwägen. Die zentrale Fragestellung lautet: Ist es sinnvoll mehr Parameter beziehungsweise ein komplexeres Modell zu verwenden oder reichen weniger Parameter schon aus um die Daten relativ gut zu erklären. In konkreter Formel sieht das Kriterium folgendermaßen aus:

$$AIC = T \left(\ln \left(\frac{\sum_{t=1}^T (y(t) - \hat{y}(t))^2}{T} \right) \right) + 2k$$

k ist die Anzahl der Parameter und T die Anzahl der Beobachtungen.

Problem bei dieser Methode ist, dass man alle möglichen Regressionen durchrechnen müsste. Dies führt schnell zu einem zu großen Rechenaufwand.

Eine gute Herangehensweise wurde in "Fast stepwise procedures of selection of variables by using AIC and BIC criteria"[AIC1] vorgeschlagen. Im Wesentlichen werden zwei Schritte gemacht: Im ersten Schritt werden alle Regressionen mit einer Variable durchgeführt und die beste ausgewählt. Ausgehend davon wird mit dieser besten Variable alle Regressionen mit 2 Variablen gerechnet. Dies wird solange gemacht, bis alle Regressanden in der Regression verwendet wurden. Hat man also zum Beispiel k erklärende Variablen so erhält man am Ende k Regressionen. Ausgehend davon werden mithilfe des AIC Kriteriums wieder die beste der k Regressionen ermittelt und man bekommt ein Ergebnis. Dies nennt man auch die Vorwärts Phase. Ein Nachteil dieser Methode ist, dass eine Variable die früh berücksichtigt wurde nicht mehr herausfällt, auch wenn sie eventuell gar nicht geeignet ist. Um diesen Schwachpunkt zu überwinden wird eine zweite Phase eingeführt. Dazu wird der Sieger der ersten Phase genommen und alle möglichen Regressionen gerechnet, wo jeweils eine Variable dazu oder wekommt. Dies macht dann also wieder k Regressionen, aus denen wieder die beste mittels AIC gewählt wird. Diese Phase bricht erst dann ab wenn es keine Verbesserung mehr gibt und wird zweite Phase genannt.

Vorteil dieses Vorgehen ist es, Parameter zu bestimmen die viel Information für die erklärenden Variable bieten. Man könnte also aus einem Pool von Parametern zu jedem Zeitpunkt jene wählen, welche den Preis am besten erklären.

Ein weiterer interessanter Aspekt ist, dass einfach zwischen der verwendeten Regression gewechselt werden kann. Der Algorithmus funktioniert für die normale lineare Regression genauso wie für Support Vector Regression, sofern die Anzahl der Parameter die Methode notwendig macht.

Die Auswertung anhand der Preisdaten für das kleinste Quadrate Kriterium befindet sich im nächsten Kapitel.

5 Prognose: Praktische Umsetzung

Im folgenden Kapitel wird die praktische Umsetzung der Prognose besprochen. Um zu testen, ob die oben gewählten Modelle geeignet für eine Prognose sind, wird mittels Backtesting das Modell direkt an den Daten angewandt.

5.1 Verwendete Bibliotheken

Die praktische Umsetzung der im vorherigen Kapitel besprochenen theoretischen Ideen erfolgte in der Software R Cran [R1], die vor allem in statistischen Anwendungen verwendet wird. In R gibt es mehrere Bibliotheken, die Support Vector Regression implementiert haben eine gute Zusammenfassung davon befindet sich in "Support Vector Machines in R"[SVR3]. In dieser Arbeit wurde die Bibliothek E1071 [R2] verwendet, welche mit *svm* und *predict* alle Funktionen enthält, die man für das Anpassen der Daten und die Prognose benötigt. Weiters sind vier Kernels und die Formulierungen 4.6 und 4.10 der Support Vector Regression unter den Namen *eps-regression* bzw. *nu-regression* implementiert. Die vier implementierten Kernels haben folgende Form:

Name	Allgemeine Form	Zu wählende Parameter	Standardwerte der Parameter
Linear	$u'v$		
Polynomial	$(\gamma u'v + coef)^{grad}$	$\gamma, coef, grad$	$\gamma = \frac{1}{k}, coef = 0, grad = 3$
Radial	$\exp(-\gamma u - v ^2)$	γ	$\gamma = \frac{1}{k}$
Sigmoid	$\tanh(\gamma u'v + coef)$	$\gamma, coef$	$\gamma = \frac{1}{k}, coef = 0$

Tabelle 5: Verwendete Parameter der Kernels

k ist dabei die Anzahl der Regressanden.

Als Input muss die Matrix mit den erklärenden Variablen, der Vektor der zu erklärenden Variable, der gewünschte Kernel sowie der Typ der Regression angegeben werden. Weitere relevante Inputmöglichkeiten für die Funktion *svm* schauen folgendermaßen aus:

Name	Erklärung
epsilon	Standardwert ist 0.1; sonst frei wählbar
cost	entspricht dem C des Strafterms, Standardwert 1
type	eps- bzw. nu-regression
class.weights	Gewichte für die einzelnen Klassen

Tabelle 6: Parameter der Kernels

Bei der Funktion *predict* muss als Input das Ergebnis von *svm* sowie die erklärenden Variablen des Prognosezeitraumes angegeben werden.

5.2 Selbst geschriebene Programme

Die Programme, welche für die Prognose verwendet wurden befinden sich in kompletter Ausführung im Anhang und in diesem Kapitel wird kurz der allgemeine Aufbau erläutert.

Als erster Schritt wurden die Daten für die Prognose in R eingelesen und in Matrizen für eine einfache Handhabung später geschrieben. Als weiterer Schritt wurde ein Vektor mit den zugehörigen Tagestypen erstellt und die Feiertage implementiert. Dazu wurden die bundesweiten Feiertage von Deutschland verwendet.

Als nächster Schritt wurden die im vorherigen Kapitel besprochenen Modelle erstellt. Dazu wurden zwei Funktionen geschrieben, die je nach Bedarf ein lineares Modell mit oder ohne exogenen Variablen sowie mit oder ohne Kreuztermen für einen gewählten Zeitraum erzeugen.

Eine weitere Funktion ist die AIC Funktion, die sich in zwei Teile aufteilt. Der erste Teil entspricht dem Forward Algorithmus und der zweite dem zweiten Teil, des schon im vorherigen Kapitel besprochenen Algorithmus. Als Input für den Forward Algorithmus benötigt man den Datensatz für den relevanten Zeitraum. Als Output bekommt man die selektierten Spalten sowie den Wert des AIC Kriteriums. Der Input des zweiten Teils entspricht dem Output des ersten, sowie dem Datensatz der relevanten Zeit. Als Output erhält man wieder die selektierten Variablen sowie den Wert des AIC Kriteriums.

Für die Prognose gibt es fünf Programme, eines für die naive, zwei weitere für die kleinste Quadrate Schätzung mit und ohne Gewichte und zwei weitere für die Support Vector Regression mit und ohne Gewichte. Die Programme laufen unabhängig voneinander, wahlweise über eine einzelne Stunde oder über alle 27 Zeitreihen und schreiben als Output, den mittleren quadratischen Fehler, den absoluten Fehler sowie die Hitrate in eine CSV-Datei für die weitere Verwendung. Dabei sind der mittlere absolute Fehler sowie der mittlere Quadratische Fehler auch tagesspezifisch verfügbar. Dabei bezeichnete *W* Wochentags, *SA* Samstags und *SO* Sonntags. Die Programme sind möglichst flexibel gestaltet um eine einfache Ergänzung der Outputs zu ermöglichen und eine genauere Erklärung befindet sich unter den Punkten der einzelnen Prognoseverfahren sowie die kompletten Programme im Anhang.

5.3 Naive Prognose

Bei der naiven Prognose wurde wie schon besprochen immer der Wert des Vortages desselben Types verwendet. Für ausgewählte Stunden sieht dies folgendermaßen aus:

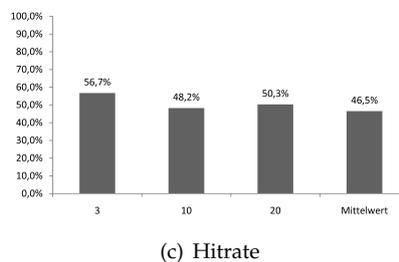
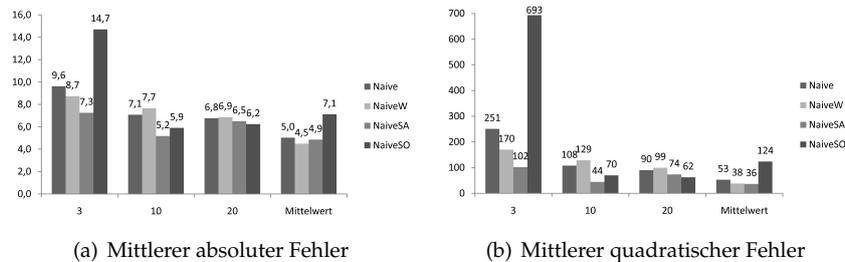


Abbildung 20: Ergebnisse der Naiven Prognose

Dabei fällt insbesondere auf, dass es bei Stunde drei am Sonntag große Ausreißer im absoluten sowie im Quadratischen Fehler gibt. Besonders gut funktioniert die naive Prognose für die Zeitreihe der mittleren Preise unter der Woche.

5.4 Kleinste Quadrate Prognose

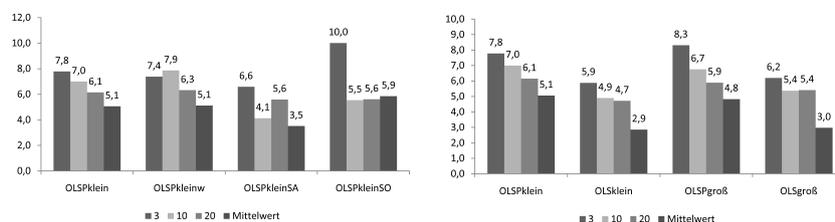
Als Basismodelle werden die vier Modelle aus den vorherigen Kapitel verwendet. Das Programm läuft über alle vier Modelle, über alle 24 einzelnen Stunden sowie über die Zeitreihe der maximalen, minimalen und mittleren Preise. Bei der kleinsten Quadrate Prognose werden grundsätzlich die Parameter für ein Jahr geschätzt um dann eine Woche zu prognostizieren. Als nächster Schritt wird dieses Fenster um eine Woche verschoben. Dadurch entstehen für den verwendeten Datensatz 49 Prognosefenster. Vor der Prognose läuft der Forward Teil des AIC Algorithmus über das erste Jahr, um ein Startregressionsmodell zu erhalten. Dieses dient dann für alle Prognosefenster als Input für den zweiten Teil des AIC Algorithmus. In jedem Prognosefenster werden folgende Schritte gemacht:

- Der Zweite Teil des AIC Algorithmus wird ausgeführt um ein "bestes Modell" aus dem zugrunde liegenden Datensatz auszuwählen.
- Die Verwendeten Parameter des "besten Modells" werden für spätere Analysezwecke abgespeichert.

- Eine Regression über das "beste Modell" wird gerechnet.
- Mittels der Werte der erklärenden Variablen wird die erklärte Variable, also die Preise für den aktuellen Zeitpunkt, prognostiziert und abgespeichert.

Nachdem die 49 Fenster gerechnet wurden, werden die Fehler in Bezug auf die tatsächlichen Preise, die Prognosefehler ausgerechnet. Diese Reihe wird als Ganzes abgespeichert sowie tagesspezifisch aufgeteilt.

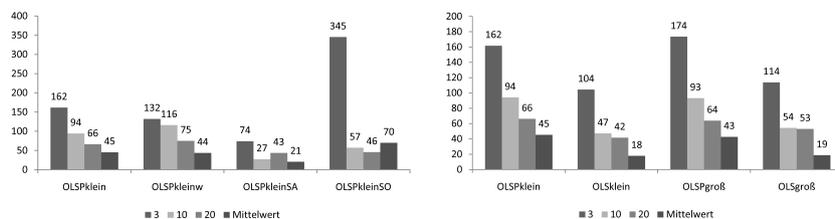
Als letzter Schritt wird der mittlere absolute Fehler, der mittlere quadratische Fehler sowie die Hitrate ausgerechnet. Die ersten Beiden werden auch tagesspezifisch ermittelt und die absolute Abweichung bezieht sich auf €/MWh. Die Ergebnisse sind in den Grafiken 21 bis 23 zu finden. Dabei bezieht sich das P im Namen immer auf das Modell ohne exogene Variablen und klein immer auf das Modell ohne Kreuzterme.



(a) Tagesspezifisch für das kleine Modell ohne exogenen Variablen

(b) Für alle 4 Modelle

Abbildung 21: OLS: Mittlere absolute Fehler



(a) Tagesspezifisch für das kleine Modell ohne exogenen Variablen

(b) Für alle 4 Modelle

Abbildung 22: OLS: Mittlere quadratische Fehler

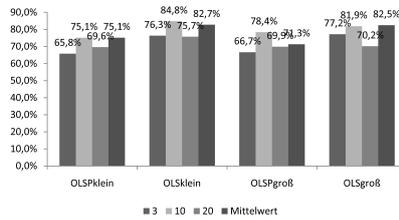


Abbildung 23: OLS: Hitrate

Bei den Ergebnissen sieht man, dass das kleine Modell mit exogenen Variablen am besten funktioniert, wobei auch das große Modell mit exogenen Variablen sehr gute Ergebnisse liefert. Weiters fällt wieder auf, dass vor allem zur Stunde drei am Sonntag das Modell keine guten Ergebnisse liefert. Dies sieht man auch, wenn man sich die Varianz der Residuen ansieht. Dabei liegt der Wert von Sonntag immer gut 2,5 mal über den Wert von der gesamten Zeitreihe zur selben Stunde. Nachdem bei den Sonntagen auch die Feiertage inkludiert sind könnte man sich für weitere Modelle überlegen diese Struktur nochmals genauer zu analysieren und in weiterer Folge eventuell ein besseres Feiertagsmodell einzubauen.

Weiters wurde noch angeschaut, welche erklärenden Variablen durch das AIC Kriterium ausgewählt werden. Die Ergebnisse befinden sich in den Grafiken 5.4 bis 5.4. Die Zahl auf der y-Achse bezieht sich dabei immer auf den Index des Parameter θ im korrespondierenden Modell. Dabei wurde der Mittelwert über alle 24 Stunden genommen, also wie oft wird dieser Parameter im Mittel über alle Stunden ausgewählt wird. Die Indizes sind nochmal in den Tabellen jeweils neben/über den Grafiken zusammengefasst. Der mittlere Wert bezieht sich darauf, wieviel Prozent der Variablen im Durchschnitt ausgewählt werden.

δ_{SASO}	δ_{SO}	Y_T	Y_T	Y_7
θ_1	θ_2	θ_3	θ_4	θ_5

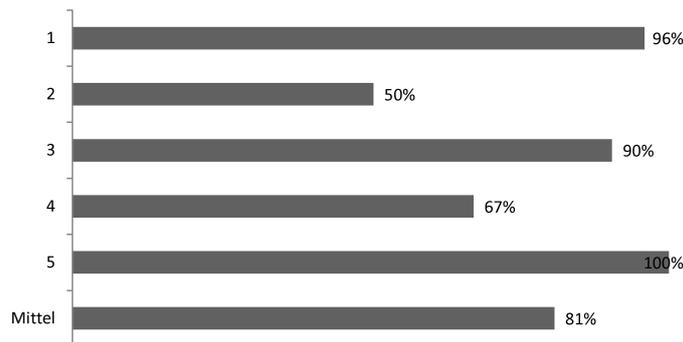


Abbildung 24: Kleines Modell ohne exogene Variablen

δ_{SASO}	δ_{SO}	\bar{Y}_T	Y_T	\bar{Y}_7	Wind	Einsatz	Temp	EXAA
θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9

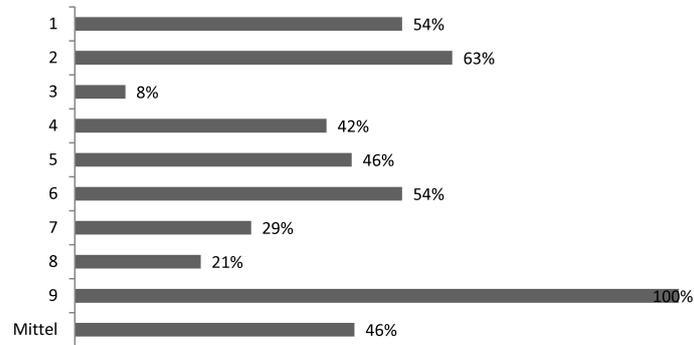


Abbildung 25: Kleines Modell mit exogenen Variablen

	1	δ_{SASO}	δ_{SO}	\bar{Y}_T	Y_T	\bar{Y}_7
1		θ_1	θ_2	θ_3	θ_4	θ_5
δ_{SASO}				θ_6	θ_7	θ_8
δ_{SO}				θ_9	θ_{10}	θ_{11}
\bar{Y}_T				θ_{12}	θ_{13}	θ_{14}
Y_T					θ_{15}	θ_{16}
\bar{Y}_7						θ_{17}

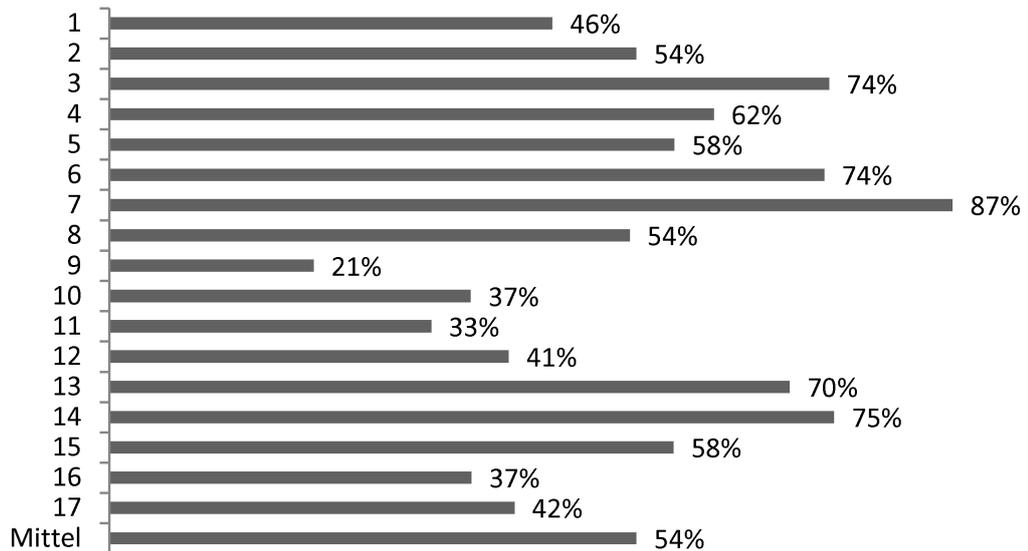


Abbildung 26: Großes Modell ohne exogene Variablen

	1	δ_{SASO}	δ_{SO}	Y_T	Y_T	Y_7	Wind	Einsatz	Temp	EXAA
1		θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9
δ_{SASO}				θ_{10}	θ_{11}	θ_{12}	θ_{13}	θ_{14}	θ_{15}	θ_{16}
δ_{SO}				θ_{17}	θ_{18}	θ_{19}	θ_{20}	θ_{21}	θ_{22}	θ_{23}
Y_T				θ_{24}	θ_{25}	θ_{26}	θ_{27}	θ_{28}	θ_{29}	θ_{30}
Y_T					θ_{31}	θ_{32}	θ_{33}	θ_{34}	θ_{35}	θ_{36}
Y_7						θ_{37}	θ_{38}	θ_{39}	θ_{40}	θ_{41}
Wind							θ_{42}	θ_{43}	θ_{44}	θ_{45}
Einsatz								θ_{46}	θ_{47}	θ_{48}
Temp									θ_{49}	θ_{50}
EXAA										θ_{51}

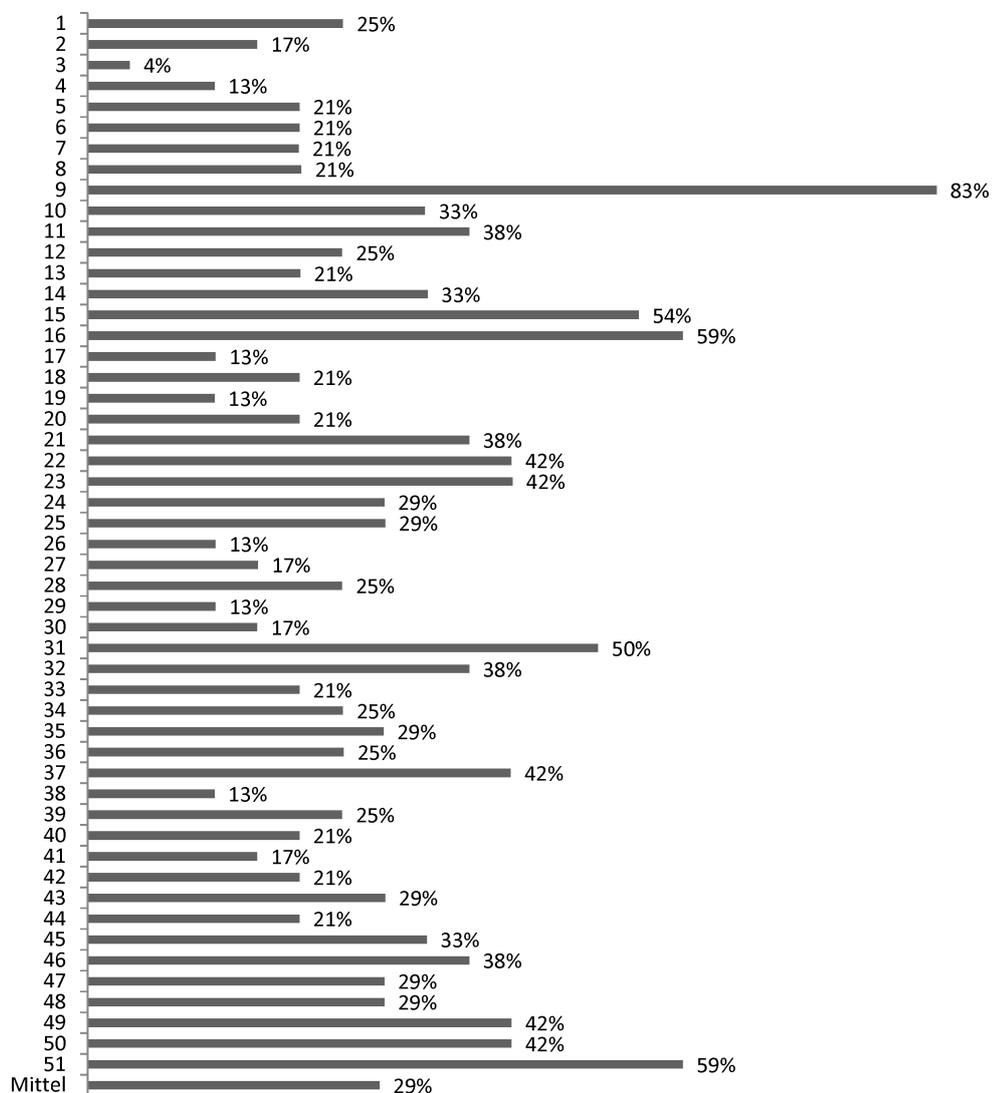


Abbildung 27: Großes Modell mit exogenen Variablen

Die Grafiken lassen vor allem auch Rückschlüsse auf die Wichtigkeit einzelner Variablen zu. Beim kleinen Modell werden vor allem die Durchschnittspreise verwendet sowie die Dummies. Beim kleinen Modell mit exogenen Variablen erkennt man, dass vor allem die Werte der Österreichischen Strombörse EXAA gewählt werden und dadurch die eigentlichen Preisdaten der EEX weniger gewählt werden. Die Dummies werden auch sehr oft selektiert und bei mehr als der Hälfte der Prognoseschritte wird auch die Windprognose verwendet. Beim großen Modell ohne exogene Variablen sind die fünf meistgewählten Variablen ist immer eine Preisvariable zusammen mit einen der Dummies, wobei keiner doppelt vorkommt, dabei sowie zwei verschiedene Kreuzterme der alten Preise. Beim großen Modell mit exogenen Variablen fällt vor allem bei den fünf meistgewählten Parametern auf, das die alten Preisdaten der EEX wieder durch die EXAA ersetzt werden und die eigentlichen Preise nur noch als quadratischer Term auftauchen. Bei den anderen exogenen Variablen wird weiters noch die Temperaturprognose mit einen Wochenenddummy ausgewählt. Dieser dürfte also am Wochenende noch zusätzliche Information liefern.

Ein alternativer Ansatz wäre noch statt eines verschobenen Fensters ein wachsendes zu nehmen. Also jeweils die alten Daten im Modell zu behalten und somit einen immer größeren Datensatz zu haben. Als letzter Schritt wurde noch die Autokorrelation der Fehler angeschaut. Das Ergebnis zeigt, dass das Modell die Struktur zufriedenstellend erfasst.

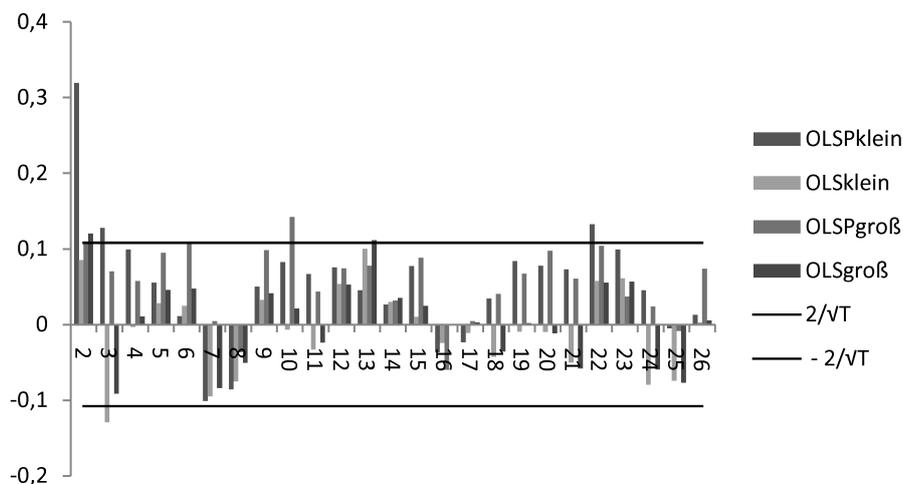


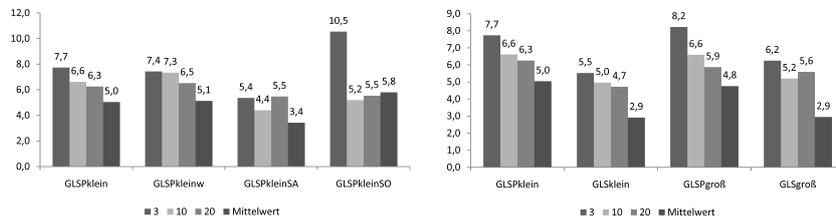
Abbildung 28: ACF der Prognosefehler im OLS Modell

5.5 Kleinste Quadrate Prognose mit Gewichten

Konkret sieht dieses zweiphasige Schätzproblem im verallgemeinerten Regressionsmodell folgendermaßen aus: Im ersten Schritt wird eine klassische kleinste Quadrate Regression mit AIC Kriterium gerechnet wie sie schon im vorherigen Abschnitt besprochen worden ist. Als nächster Schritt werden die Residuenquadrate der Regression mittels Tagesdummies erklärt was folgendermaßen aussieht:

$$res^2 = \beta_1 \delta_{WT} + \beta_2 \delta_{Sa} + \beta_3 \delta_{So} \quad (5.1)$$

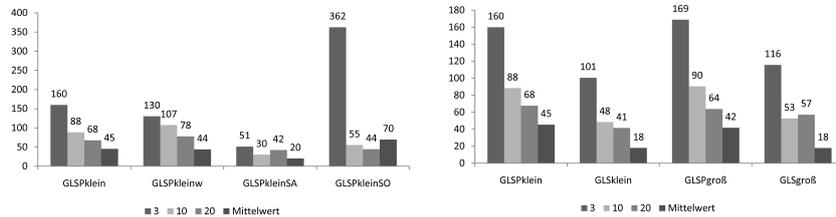
Mit *WT* ist Wochentag, *Sa* ist Samstag und *So* ist Sonntag. Damit versucht man ein Tagesspezifisches Varianzniveau herauszufiltern um dieses dann als Schätzer für die unterschiedlichen Varianzen zu verwenden. Als nächster Schritt wird der Vektor *y* sowie die Matrix *X* komponentenweise durch die Wurzel der Varianz des Tagestyp zum Zeitpunkt der erklärten Variable, *y* dividiert. Danach wird mit den modifizierten Matrizen die Prognose (wieder unter Verwendung des AIC-Kriteriums) gerechnet. Als letzter Schritt werden die Prognostizierten Werte zurückgerechnet indem man mit der Wurzel der Tagesspezifischen Varianz multipliziert.



(a) Tagesspezifisch für das kleine Modell ohne exogenen Variablen

(b) Für alle 4 Modelle

Abbildung 29: GLS: Mittlere absolute Fehler



(a) Tagesspezifisch für das kleine Modell ohne exogenen Variablen

(b) Für alle 4 Modelle

Abbildung 30: GLS: Mittlere quadratische Fehler

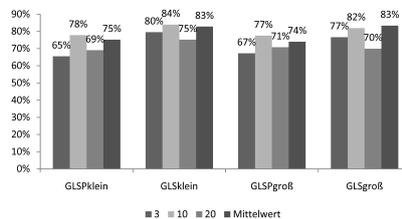


Abbildung 31: GLS: Hitrate

Bei den Ergebnissen sieht man eine sehr ähnliche Struktur zu jenen der der

Kleinsten Quadrate Prognose ohne Gewichten. Das große Modell mit exogenen Variablen ist eine Spur besser als ohne Gewichte.

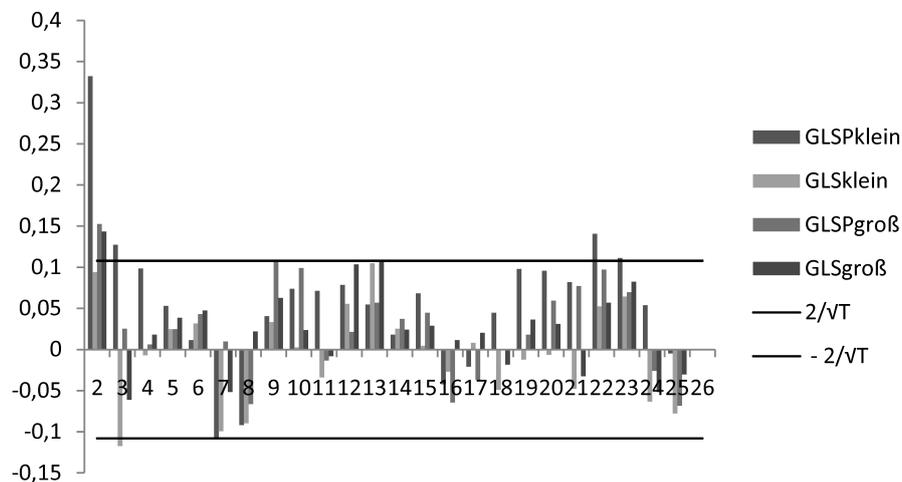


Abbildung 32: ACF der Prognosefehler im GLS Modell

Die Autokorrelation der Fehler wurde bei den nicht Rücktransformierten Variablen angeschaut, da nur in diesen Schritt sichtbar ist ob das Modell die Struktur der Daten richtig erfasst. Das Ergebnis ist ähnlich zufriedenstellend wie schon bei der Prognose ohne Gewichte.

5.6 Support Vector Regression

Bei der Prognose in der Support Vector Regression wird gleich vorgegangen wie bei der kleinsten Quadrate Schätzung. Es werden die Schätzer für die ersten 365 Tage ermittelt und in weiterer Folge die Prognose für die darauffolgenden sieben Tage gerechnet. Danach wird das Fenster um eine Woche verschoben, um dann wie bei der kleinsten Quadrate Schätzung auf 49 Intervalle zu kommen. Verwendet werden dabei nur die beiden kleinen Modelle, also eines mit und eines ohne exogenen Variablen.

5.6.1 Parameter der Kernels

Als erster Schritt in der Support Vector Regression werden die Parameter der vier Kernels angepasst. Dazu wird ein erster Durchlauf mit den Standardwerten ($\epsilon = 0,1$ und $C = 1$) gerechnet und dann Schritt für Schritt die einzelnen Parameter der Funktionen (siehe Abschnitt verwendet Bibliotheken für die Parameter der einzelnen Kernels) zu ändern um einen optimalen Wert zu finden. Dieser Schritt ist nicht unbedingt die beste Herangehensweise, aber sicherlich ein guter erster Ansatz um auszutesten wie sensibel die Ergebnisse auf Veränderung der einzelnen Parameter reagieren. Dazu wurde kein eigener Algorithmus geschrieben, sondern nur iterativ die einzelnen Werte verändert. Weiters wurden nicht alle Stunden gerechnet, sondern die Zeitreihe der mittleren Preise verwendet. Als zugrunde liegendes Modell wurde jenes ohne exogene Parameter und Kreuztermen verwendet.

Beim linearen Kernel gibt es keine wählbaren Parameter und das Ergebnis befindet sich in Tabelle 7, wobei sich 1 und 2 auf *eps-regression* und *nu-regression* beziehen.

	Mittlerer Absoluter Fehler	Mittlerer Quadratischer Fehler	Hitrate
SVRPLinear1	5,203	62,081	72,8%
SVRPLinear2	5,204	62,128	72,5%

Tabelle 7: SVR: Ergebnis mit linearem Kernel

Beim polynomialen Kernel gibt es schon mehr Parameter, die man an der Funktion verändern kann. Die allgemeine Form des Kernels ist folgende:

$$(\gamma u'v + coef)^{grad}$$

Es gibt also drei Parameter, die man verändern kann, γ , *coef* und *grad*. Der Skalierungsfaktor γ wurde auf seinem Standardwert belassen. Bei *coef* und *grad* wurden einige Werte ausprobiert. In Tabelle 8 befinden sich die Ergebnisse durch Anpassen des Parameters *coef* und in Tabelle 9 jene durch Anpassen des Parameters *grad*. Für die weitere Prognose wurden die Parameter mit den besten Ergebnis gewählt also *coef* = 0.25 und *grad* = 3 (der Standardwert).

		Standard (0)	0.25	0.5	1	2.5	5	10
SVRPPoly1	Mittlerer Absoluter Fehler	6,15	5,27	5,27	5,27	5,32	5,36	5,41
SVRPPoly2	Fehler	6,35	5,36	5,34	5,41	5,44	5,46	5,48
SVRPPoly1	Mittlerer Quadratischer Fehler	75,98	64,16	65,10	66,54	69,18	71,30	74,60
SVRPPoly2	Fehler	78,85	65,64	66,47	68,89	70,32	71,65	73,24
SVRPPoly1	Hitrate	72,8%	71,3%	71,6%	72,2%	71,6%	71,6%	71,3%
SVRPPoly2		71,3%	72,5%	71,9%	71,6%	71,6%	71,6%	71,1%

Tabelle 8: SVR: Ergebnis mit polynomialen Kernel und Veränderung des Parameter coef

		Standard (3)	0.25	0.5	5
SVRPPoly1	Mittlerer Absoluter Fehler	6,15	16,66	16,66	7,08
SVRPPoly2	Fehler	6,35	17,53	17,53	7,43
SVRPPoly1	Mittlerer Quadratischer Fehler	75,98	401,29	401,29	101,41
SVRPPoly2	Fehler	78,85	415,65	415,65	109,75
SVRPPoly1	Hitrate	72,8%	50,3%	50,3%	68,7%
SVRPPoly2		71,3%	49,1%	49,1%	69,0%

Tabelle 9: SVR: Ergebnis mit polynomialen Kernel und Veränderung des Parameter grad

Der radiale Kernel hat die Form

$$\exp(-\gamma |u - v|^2)$$

wobei der einzige Parameter γ ist und dieser auch hier nicht verändert wird. Die Ergebnisse der Prognose sind in Tabelle 10 zu finden.

Beim sigmoiden Kernel gibt es wieder mehr Parameter, die man an der Funktion verändern kann. Die allgemeine Form des Kernels ist folgende:

$$\tanh(\gamma u'v + coef)$$

Der Parameter γ wird auch hier nicht verändert. Der Parameter *coef* wird angepasst, wobei schnell erkennbar ist, dass dieser Kernel für die Daten eher ungeeignet ist und sehr große Fehler liefert. Die Ergebnisse befinden sich in Tabelle 11 und der Kernel wird in weiteren Prognosen nicht mehr berücksichtigt.

	Mittlerer Absoluter Fehler	Mittlerer Quadratischer Fehler	Hitrate
SVRPRadial1	5,252	63,482	69,6%
SVRPRadial2	5,347	64,941	68,7%

Tabelle 10: SVR: Ergebnis mit radialen Kernel

		Standard (0)	0.25	0.5	1	2.5	5	10
SVRPSigmoid1	Mittlerer Absoluter Fehler	177,58	218,10	257,61	367,50	110,46	15,10	16,66
SVRPSigmoid2	Fehler	145,66	173,38	189,53	292,88	98,10	16,13	17,53
SVRPSigmoid1	Mittlerer Quadratischer Fehler	71.490,32	82.548,90	98.901,64	258.167,50	44.246,78	322,52	401,28
SVRPSigmoid2	Fehler	36.453,63	48.015,72	50.202,79	132.953,97	29.777,27	345,25	415,65
SVRPSigmoid1	Hitrate	42,7%	43,9%	55,3%	50,3%	63,7%	52,3%	50,3%
SVRPSigmoid2		51,2%	51,5%	59,9%	49,1%	61,1%	50,0%	49,1%

Tabelle 11: SVR: Ergebnis mit sigmoiden Kernel und Veränderung des Parameter coef

5.6.2 Parameter der eps- und nu-regression

Nachdem die Parameter der Kernels angepasst wurden, kann man noch jene bei der *eps-regression* und *nu-regression* ändern. Dazu wurde der Parameter *cost* und *epsilon* verändert. Die Ergebnisse für alle Kernels befinden sich in den Tabellen 12 bis 14, wobei die Standardwerte bei *cost* = 1 und *epsilon* = 0.1 liegen

	Normal	epsilon					cost	
		0.2	0.5	1	1.5	2	10	2
SVRPLinear1	5,20	5,17	8,15	7,77	7,87	7,53	7,53	7,53
SVRPPoly1	6,15	6,28	9,31	7,79	14,86	8,16	7,54	8,16
SVRPRadial1	5,25	5,32	9,00	8,03	16,42	6,94	6,97	6,94
SVRPSigmoid1	177,58	177,57	240,71	140,60	37,13	479,66	2.399,69	479,66
SVRPLinear2	5,20	5,20	7,60	6,83	5,20	7,60	7,60	7,60
SVRPPoly2	6,35	6,35	8,84	6,98	6,35	8,33	7,58	8,33
SVRPRadial2	5,35	5,35	7,16	7,28	5,35	7,04	7,05	7,04
SVRPSigmoid2	145,66	145,66	196,84	118,11	145,66	392,35	1.960,42	392,35

Tabelle 12: SVR: Mittlere absolute Abweichung bei Veränderung der Parameter epsilon und cost

	Normal	epsilon					cost	
		0.2	0.5	1	1.5	2	10	2
SVRPLinear1	62,08	61,85	118,94	217,82	109,17	112,57	112,56	112,57
SVRPPoly1	75,98	78,37	149,94	227,70	293,22	126,28	112,15	126,28
SVRPRadial1	63,48	64,16	133,65	227,51	350,61	104,83	106,29	104,83
SVRPSigmoid1	71.490,32	71.492,10	131.562,03	43.530,14	4.922,76	524.683,23	13.182.931,36	524.683,23
SVRPLinear2	62,13	62,13	114,13	217,15	62,13	114,07	114,06	114,07
SVRPPoly2	78,85	78,85	142,15	208,51	78,85	129,42	112,71	129,42
SVRPRadial2	64,94	64,94	107,66	220,34	64,94	106,20	107,50	106,20
SVRPSigmoid2	36.453,63	36.453,63	68.297,54	22.424,02	36.453,63	272.854,65	6.813.234,13	272.854,65

Tabelle 13: SVR: Mittlere quadratische Abweichung bei Veränderung der Parameter epsilon und cost

	Normal	epsilon					cost	
		0.2	0.5	1	1.5	2	10	2
SVRPLinear1	72,8%	73,4%	70,8%	67,5%	65,2%	76,6%	76,6%	76,6%
SVRPPoly1	72,8%	71,6%	72,8%	65,5%	57,9%	74,0%	76,3%	74,0%
SVRPRadial1	69,6%	71,6%	63,7%	64,0%	49,1%	75,1%	74,3%	75,1%
SVRPSigmoid1	42,7%	42,4%	45,0%	50,3%	49,7%	52,6%	50,9%	52,6%
SVRPLinear2	72,5%	72,5%	76,3%	65,5%	72,5%	76,3%	76,3%	76,3%
SVRPPoly2	71,3%	71,3%	71,9%	67,0%	71,3%	72,5%	74,6%	72,5%
SVRPRadial2	68,7%	68,7%	73,7%	66,1%	68,7%	75,4%	74,9%	75,4%
SVRPSigmoid2	51,2%	51,2%	48,5%	50,0%	51,2%	51,8%	52,3%	51,8%

Tabelle 14: SVR: Hitrate bei Veränderung der Parameter epsilon und cost

Es werden also die Standardwerte beibehalten und man erhält abschließend folgende Parameterauswahl, die ab jetzt auch in allen weiteren Berechnungen verwendet wird:

Name	Allgemeine Form
Linear	$u'v$
Polynomial	$(\frac{1}{k}u'v + 0.25)^3$
Radial	$\exp(-\frac{1}{k} u-v ^2)$
epsilon	0.1
cost	1

Tabelle 15: Form der Kernels

Diese Argumentation müsste jetzt auch für das Modell mit exogenen Variablen geführt werden, sowie für die Zeitreihen der einzelnen Stunden. Rechnet man mit den gefundenen Parametern aus dem Modell ohne exogenen Parametern, so erhält man schon eine deutliche Verbesserung. Eine genauere Parameteranpassung wird nicht mehr vorgenommen. Weiters soll angemerkt werden, dass es mittlerweile eine eigene Funktion *tune.svm* gibt, die die Parameter optimiert. Nach Ausprobieren dieser Funktion stellte sich heraus, dass die Ergebnisse nicht wesentlich von den hier gefundenen abweichen und damit wurde dieser Punkt nicht mehr nachträglich geändert.

5.6.3 Ergebnis

Abschließend wurde noch mit den angepassten Parametern über alle Stunden gerechnet und das Ergebnis wird in den Abbildungen 33 bis 35 präsentiert. Der sigmoide Kernel wurde nicht mehr in die Abbildung hineingenommen nachdem die Ergebnisse deutlich schlechter als die naive Prognose ausfallen. Sonst liefern vor allem die Ergebnisse mit linearem Kernel und mit exogenen Variablen gute Resultate über alle Kriterien. Bei der quadratischen Abweichung merkt man jedoch nur wenig Unterschiede zwischen den verschiedenen Kernels. Im Allgemeinen ist der lineare ein wenig besser als der polynomiale und als der radiale Kernel. Des weiteren funktioniert die ϵ -Regression ein wenig besser als die ν -Regression.

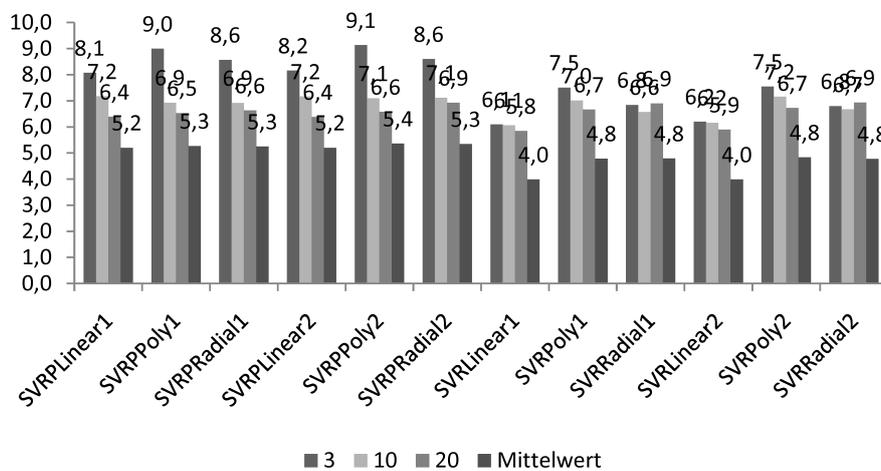


Abbildung 33: SVR Ergebnis: Mittlere absolute Abweichung

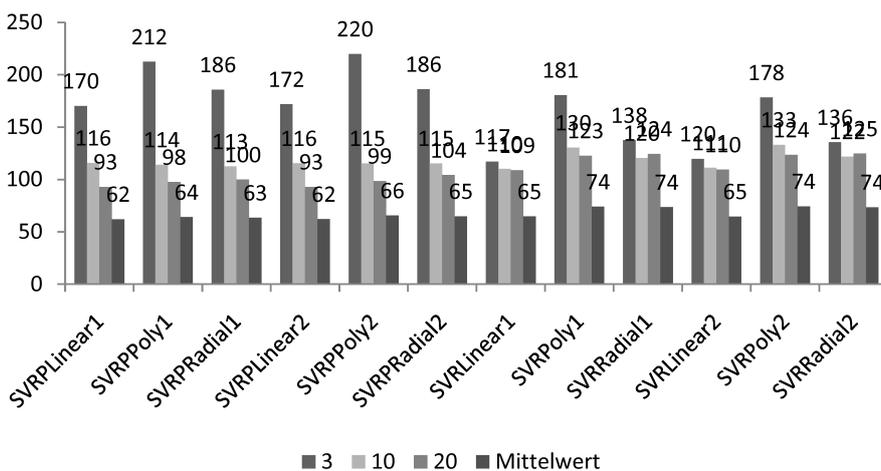


Abbildung 34: SVR Ergebnis: Mittlere quadratische Abweichung

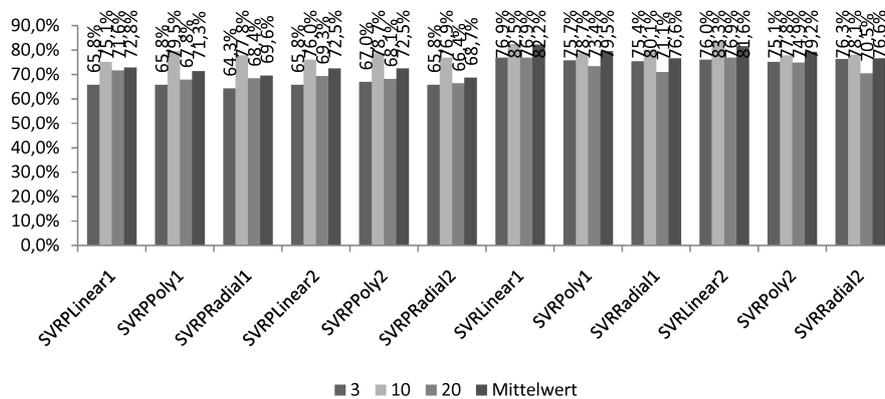


Abbildung 35: SVR Ergebnis: Hitrate

5.7 Support Vector Regression mit Gewichten

Um die tagesabhängige Varianzstruktur auch in der Support Vector Regression zu berücksichtigen wird folgendermaßen vorgegangen: Es wird wie bei der vorhin besprochenen Support Vector Regression eine Regression mit dem respektiven Modell und Kernel gerechnet. Im nächsten Schritt werden die tagesspezifischen Varianzen mit Formel 5.1 und normaler kleinster Quadrate Regression gerechnet und die Daten anschließend wie beim verallgemeinerten Regressionsmodell transformiert. Als nächstes wird die Support Vector Regression mit den modifizierten Daten gerechnet, die Prognose gemacht und das Ganze zurückgerechnet. Die Einbeziehung der Gewichte hat auch bei der Support Vector Regression nur einen kleinen Einfluss auf die Ergebnisse.

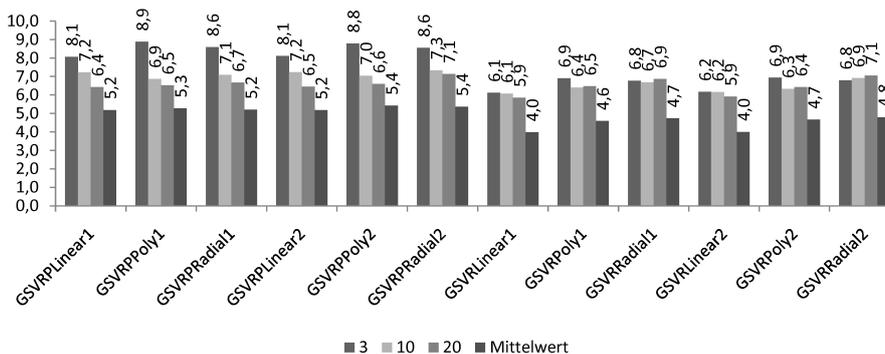


Abbildung 36: SVRG: Mittlere absolute Abweichung

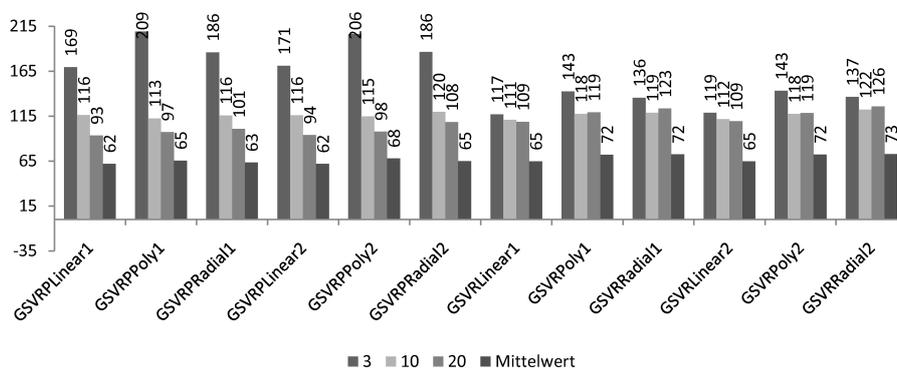


Abbildung 37: SVRG: Mittlere quadratische Abweichung

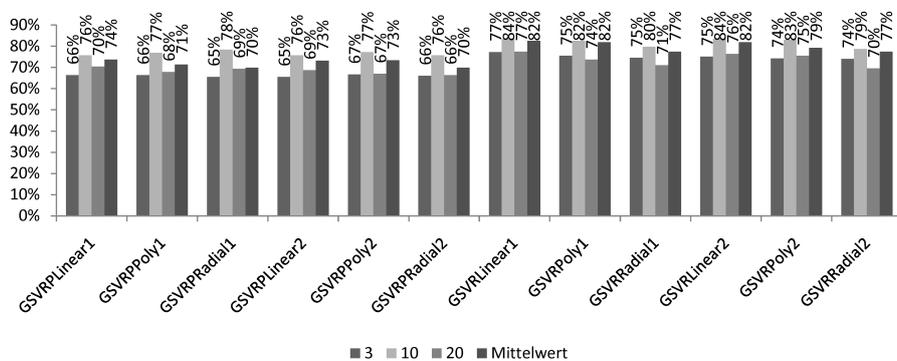


Abbildung 38: SVRG: Hitrate

5.8 Vergleich der Ergebnisse

Als abschließende Frage der verschiedenen Prognoseansätze stellt sich natürlich die Frage, welches Kriterium sich am besten für die Prognose von Strompreisen eignet. Dazu wurden die Ergebnisse der Zeitreihe der mittleren Preise zusammengefasst und sie befinden sich in den Abbildungen 39 bis 41. Es zeigt sich dabei, dass die Naive Prognose von vielen anderen Ansätzen in allen drei Kriterien geschlagen wird. Vor allem bei der Hitrate kann die Naive Prognose nicht mithalten. Die besten Ergebnisse insgesamt liefern die kleinen und großen Modelle mit Einbeziehung der exogenen Variablen und unter Verwendung von kleinster Quadrate Prognose mit und ohne Gewichten. Im quadratischen Fehler liegt die Support Vector Regression hinter den Ergebnissen der Naiven Prognose. Beim absoluten Fehler liefert die Support Vector Regression bei den Modellen mit exogenen Variablen bessere Ergebnisse als die naive Prognose.

5.9 Veränderung des Ergebnis durch Verwendung des AIC Kriteriums

Bei der Methode der kleinsten Quadrate sowie beim Ansatz mit tagesspezifischen Gewichten wurde das AIC Kriterium verwendet um eine möglichst gute Erklärung zu erhalten. Es stellte sich die Frage, inwiefern das Kriterium seinen Ansprüchen gerecht wird. Dazu wurde die Prognose über die Zeitreihe der mittleren Preise einmal mit und einmal ohne AIC-Kriterium gerechnet. Die Ergebnisse befinden sich in den Abbildungen 42 bis 42 und es zeigt sich, dass vor allem bei den großen Modellen das AIC Kriterium eine deutliche Verbesserung bringt.

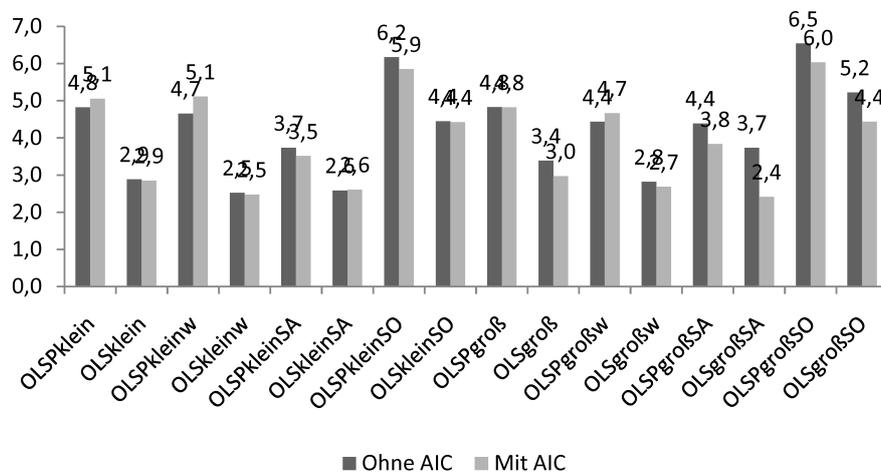


Abbildung 42: Einfluss AIC: Mittlerer Absoluter Fehler

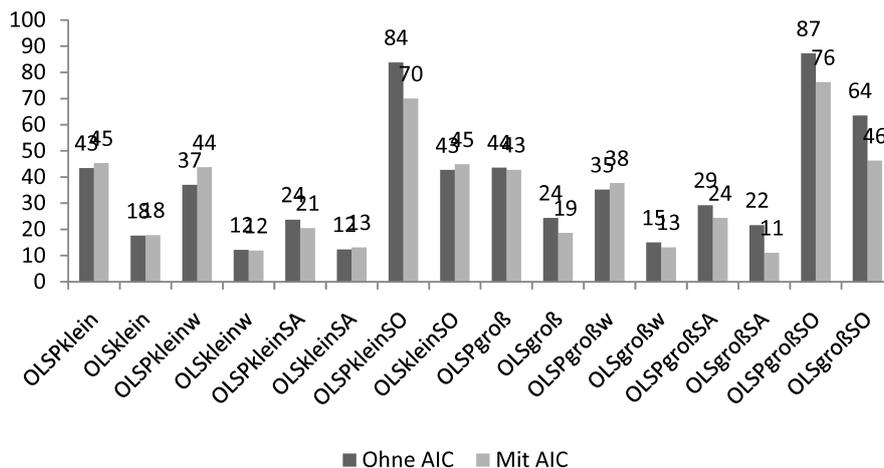


Abbildung 43: Einfluss AIC: Mittlerer Quadratischer Fehler

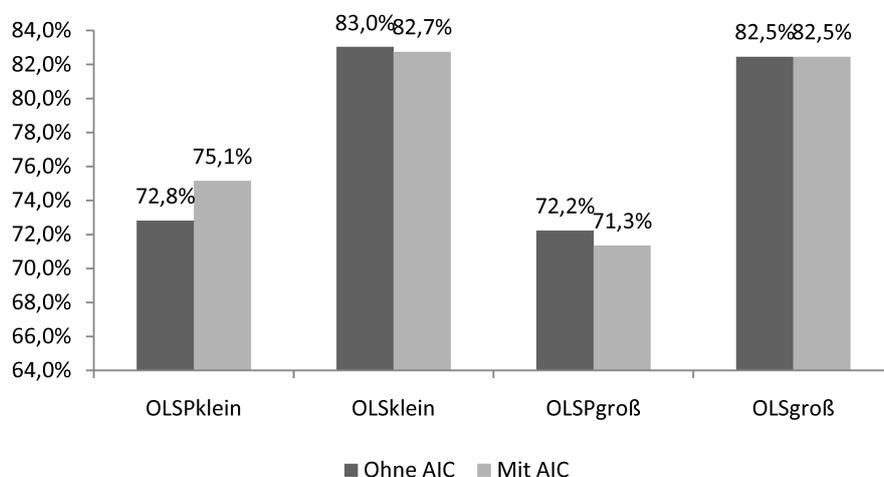


Abbildung 44: Einfluss AIC: Hitrate

5.10 Einfluss der einzelnen exogenen Variablen

In der Prognose der Strompreise in den Modellen mit exogenen Parametern gehen vier verschiedene Parameter ein, Wind, Temperatur, Lastprognosen sowie der Preis der Österreichischen Börse. Nun ist es interessant herauszufiltern wieviel Einfluss das Dazunehmen von einer exogenen Variable auf den Preis hat. Dazu wird jeweils immer die Zeitreihe der mittleren Preise mit dem kleinen Modell ohne exogene Parameter verwendet und abwechselnd immer eine Variable dazugenommen. Das AIC Kriterium wird dabei nicht verwendet. Dies wird zuerst für die kleinste Quadrate Methode gerechnet, wobei der Einfluss des Parameters direkt sowie der Einfluss des Parameters zusammen mit dessen quadratischen Term ausgewertet wird. Weiters wird der Einfluss der einzelnen Terme auch in der Support Vector Regression gerechnet, wobei hier natürlich keine quadratischen Terme verwendet werden.

		Wind	Temp	Einsatz	EXXA
Nur lineare Terme	MAD	5,37%	-0,69%	-3,13%	39,29%
	MSD	8,32%	-0,25%	-3,41%	57,56%
	Hitrate	3,19%	-0,40%	0,00%	14,46%
Lineare und quadratische Terme	MAD	4,87%	0,44%	-3,47%	38,74%
	MSD	11,16%	1,79%	-4,47%	57,09%
	Hitrate	4,02%	1,61%	-1,20%	14,46%

Abbildung 45: Verbesserung durch die einzelnen exogenen Variablen bei der kleinsten Quadrate Schätzung

Bei der kleinsten Quadrate Prognose bringt im Allgemeinen die EXAA die meiste Verbesserung. Als zweites kommt die Windprognose und danach Temperatur und Einsatzprognosen die aber sogar teilweise einen negativen Einfluss haben.

	Wind	Temp	Last	EXAA
SVRLinear1	-7%	-13%	-16%	23%
SVRPoly1	-9%	-13%	-20%	16%
SVRRadial1	-9%	-12%	-16%	15%
SVRLinear2	-8%	-14%	-17%	23%
SVRPoly2	-8%	-13%	-19%	17%
SVRRadial2	-9%	-14%	-16%	16%

(a) Mittlerer absoluter Fehler

	Wind	Temp	Last	EXAA
SVRLinear1	-37%	-44%	-47%	-5%
SVRPoly1	-37%	-41%	-55%	-9%
SVRRadial1	-38%	-39%	-49%	-12%
SVRLinear2	-37%	-46%	-48%	-5%
SVRPoly2	-34%	-39%	-53%	-7%
SVRRadial2	-37%	-40%	-47%	-11%

(b) Mittlerer quadratischer Fehler

	Wind	Temp	Last	EXAA
SVRLinear1	2%	-1%	-3%	14%
SVRPoly1	4%	-2%	-2%	16%
SVRRadial1	3%	-1%	-1%	15%
SVRLinear2	3%	-2%	-2%	15%
SVRPoly2	0%	-2%	-2%	12%
SVRRadial2	5%	1%	0%	16%

(c) Hitrate

Abbildung 46: Verbesserung durch die einzelnen exogenen Variablen bei der Support Vector Regression

Bei der Support Vector Regression ist der Einfluss um einiges kleiner wie bei der kleinsten Quadrate Prognose und die meisten Werte werden sogar negativ.

6 Conclusio

Die Struktur der Spotpreise für Strom legte es nahe, ein tagestypabhängiges Modell zu verwenden. Dazu wurden zusammen mit den exogenen Daten verschiedene erklärende Variablen definiert und verschiedene Modelle ange-dacht. In weiterer Folge wurden fünf verschiedene Prognoseansätze ausprobiert wobei sich heraus stellte, dass die kleinste Quadrate Prognose (mit und ohne Gewichten) schwer zu schlagen ist und für große Modelle eine Variablen-selektion mittels AIC Kriterium unumgänglich wird. Die Support Vector Regression kann von der Größenordnung mithalten, schlägt aber in keinem Ansatz die kleinste Quadrate Prognose. Dabei soll festgehalten werden, dass die Parameter der Kernels, abgesehen von der Zeitreihe der mittleren Preise, noch einen Verbesserungsspielraum haben und auch noch andere Kernels ausgetestet werden könnten. Bei der Untersuchung des Einflusses der einzel-

nen exogenen Variablen stellte sich heraus, dass die Werte der österreichischen Strombörse eine wesentliche Verbesserung bringen und die Werte der Windprognose auch eine exaktere Schätzung bewirken.

Appendix

Literatur

- [BD] Peter J. Brockwell, Richard A. Davis. 1991. "Time Series: Theory and Methods". Springer Series in Statistics
- [ÖK2] M. Deistler, W. Scherrer 1992. "Econometrics II". Lectures given at CER-GE Prague
- [EEX1] http://www.eex.com/de/document/4423/Einführung_Börsenhandel_Release_01B.pdf. Zugriff: 11. März 2010
- [EEX2] <http://www.eex.com/de/EEX/EEX/Unternehmensstruktur>. Zugriff: 11. März 2010
- [EEX3] http://static.epexspot.com/document/7087/20100101_EPEX_Spot_Rules_&_Regulations_DE.zip. Zugriff: 30. April 2010
- [Wind] EuroWind GmbH. <http://www.windprognose.de/Ueberuns/impressum.php>. Zugriff: 1. Mai 2010
- [EXAA1] http://www.exaa.at/market/trade_results/aggregated_curves/. Zugriff: 25. März 2010
- [EXAA2] http://www.exaa.at/static/cms/sites/exaa.at/media/downloads/EXAA_Folder_Power_2010.pdf. Zugriff: 11. März 2010
- [AIC1] H. An, L. Gu. 1989. "Fast stepwise procedures of selection of variables by using AIC and BIC criteria". Acta Mathematicae Applicatae Sinica 5, 60-67
- [SVR1] Alex J. Smola, Bernhard Schölkopf. "A Tutorial on Support Vector Regression". September 30, 2008
- [SVR2] V. Vapnik. "The nature of Statistical Learning Theory". Springer, New York 1995
- [SVR3] A. Kratochvíl, D. Meyer, K. Hornik. "Support Vector Machines in R". Journal of Statistical Software, April 2006, Vol 15, Issue 9
- [R1] <http://cran.r-project.org/>. Zugriff: 10.11.2010
- [R2] <http://cran.r-project.org/web/packages/e1071/index.html>. Zugriff: 10.11.2010

Abbildungsverzeichnis

1	Beispiel eines Stundenkontrakts	3
2	Beispiel einer aggregierten Angebots- bzw. Nachfragekurve an der EXAA 25. März 2010 Stunde 18	5
3	Ausreißer und negative Werte	6
4	Gesamte Zeitreihe von 14.1.2008 bis 27.1.2008	7
5	Zeitreihe einzelner Stunden von 1.1.2008 bis 31.12.2008	8
6	Zeitreihe der mittleren, minimalen und maximalen Tageswerte von 1.1.2008 bis 31.12.2008	8
7	ACF der gesamten Zeitreihe	10
8	PACF der gesamten Zeitreihe	11
9	ACF einzelner Stunden	11
10	PACF einzelner Stunden	12
11	Periodogramm der gesamten Zeitreihe	13
12	Periodogramm einzelner Stunden	13
13	Intensität des Periodogramms in Abhängigkeit der gewählten Stunde	14
14	Mittelwerte(y-Achse) der einzelnen Stunden(x-Achse) abhängig vom Tagestyp	16
15	Varianzen(y-Achse) der einzelnen Stunden(x-Achse) abhängig vom Tagestyp	16
16	Mittelwerte abhängig von der Stunde mit 15 gleitenden Fenstern	17
17	Varianzen abhängig von der Stunde mit 15 gleitenden Fenstern	17
18	Windprognose vom 7.1.2008 bis 13.1.2008	18
19	Naive Prognose	23
20	Ergebnisse der Naiven Prognose	31
21	OLS: Mittlere absolute Fehler	32
22	OLS: Mittlere quadratische Fehler	32
23	OLS: Hitrate	33
24	Kleines Modell ohne exogene Variablen	33
25	Kleines Modell mit exogenen Variablen	34
26	Großes Modell ohne exogene Variablen	34
27	Großes Modell mit exogenen Variablen	35
28	ACF der Prognosefehler im OLS Modell	36
29	GLS: Mittlere absolute Fehler	37
30	GLS: Mittlere quadratische Fehler	37
31	GLS: Hitrate	37
32	ACF der Prognosefehler im GLS Modell	38
33	SVR Ergebnis: Mittlere absolute Abweichung	42
34	SVR Ergebnis: Mittlere quadratische Abweichung	42
35	SVR Ergebnis: Hitrate	43
36	SVRG: Mittlere absolute Abweichung	43
37	SVRG: Mittlere quadratische Abweichung	44
38	SVRG: Hitrate	44
39	Gesamtergebnis: Mittlerer absoluter Fehler	45
40	Gesamtergebnis: Mittlerer quadratischer Fehler	45
41	Gesamtergebnis: Hitrate	45
42	Einfluss AIC: Mittlerer Absoluter Fehler	46
43	Einfluss AIC: Mittlerer Quadratischer Fehler	46

44	Einfluss AIC: Hitrate	47
45	Verbesserung durch die einzelnen exogenen Variablen bei der kleinsten Quadrate Schätzung	47
46	Verbesserung durch die einzelnen exogenen Variablen bei der Support Vector Regression	48

Tabellenverzeichnis

1	Wertepaare	3
2	Ergebnisse der Trendregression	9
3	Erklärende Variablen der Preise	20
4	Exogene Variablen	21
5	Verwendete Parameter der Kernels	29
6	Parameter der Kernels	30
7	SVR: Ergebnis mit linearem Kernel	39
8	SVR: Ergebnis mit polynomialen Kernel und Veränderung des Parameter coef	39
9	SVR: Ergebnis mit polynomialen Kernel und Veränderung des Parameter grad	39
10	SVR: Ergebnis mit radialen Kernel	40
11	SVR: Ergebnis mit sigmoiden Kernel und Veränderung des Parameter coef	40
12	SVR: Mittlere absolute Abweichung bei Veränderung der Parameter epsilon und cost	40
13	SVR: Mittlere quadratische Abweichung bei Veränderung der Parameter epsilon und cost	41
14	SVR: Hitrate bei Veränderung der Parameter epsilon und cost	41
15	Form der Kernels	41

A Quellcodes

A.1 Daten einlesen

```
#####  
### In richtigen Ordner wechseln  
#####  
  
setwd("C:/Users/Tina/Documents/DA/Daten")  
  
#####  
### Daten einlesen  
#####  
  
data1<-read.table("Strompreise.txt",header=TRUE,sep="\t")  
#Datensatz mit eex,exaa,six,wind  
data2<-read.table("Prognosen.txt",header=TRUE,sep="\t")  
#Datensatz mit Verfügbarkeit von Kern,Braun,Stein,Gas,Wasser,Sonstige  
und Prognose von Temperatur  
  
#####  
### Matrizen mit Daten befüllen  
#####  
  
D=730 # Anzahl der Daten  
EEX<-matrix(ncol=D,nrow=27) ###Matrizen initialisieren  
EXAA<-matrix(ncol=D,nrow=27)  
SIX<-matrix(ncol=D,nrow=27)  
Wind<-matrix(ncol=D,nrow=27)  
  
EEXg<-data1[,1]  
for (j in 0:D-1){  
EEX[,j+1]<-c(data1[((j*24)+1):((j+1)*24)],1]  
,mean(data1[((j*24)+1):((j+1)*24)],1),min(data1[((j*24)+1):((j+1)*24)],1])  
,max(data1[((j*24)+1):((j+1)*24)],1]))  
}  
for (j in 0:D-1){  
EXAA[,j+1]<-c(data1[((j*24)+1):((j+1)*24)],2]  
,mean(data1[((j*24)+1):((j+1)*24)],2),min(data1[((j*24)+1):((j+1)*24)],2])  
,max(data1[((j*24)+1):((j+1)*24)],2]))  
}  
for (j in 0:D-1){  
SIX[,j+1]<-c(data1[((j*24)+1):((j+1)*24)],3]  
,mean(data1[((j*24)+1):((j+1)*24)],3),min(data1[((j*24)+1):((j+1)*24)],3])  
,max(data1[((j*24)+1):((j+1)*24)],3]))  
}  
for (j in 0:D-1){  
Wind[,j+1]<-c(data1[((j*24)+1):((j+1)*24)],4]  
,mean(data1[((j*24)+1):((j+1)*24)],4),min(data1[((j*24)+1):((j+1)*24)],4])  
,max(data1[((j*24)+1):((j+1)*24)],4]))  
}
```

```

}

Prognose<-matrix(ncol=D,nrow=15)
for (j in 1:13){
Prognose[j,<-data2[1:D,j]
}
for (j in 8:13){ #####Fehlende Einsatzwerte werden
for (i in 1:D){ #####durch Prognosewerte ersetzt
if(is.na(Prognose[j,i])==TRUE){
Prognose[j,i]=Prognose[j-7,i]
}
}
}
for (i in 1:D){
Prognose[14,i]<-sum(Prognose[1:6,i])
Prognose[15,i]<-sum(Prognose[8:13,i])
}
Tagestyp<-c(seq(D))
for (j in 0:(D%/7-1)){
Tagestyp[((j*7)+1):((j+1)*7)]<-c(1,1,1,1,2,3,1)
}
Tagestyp[729:730]=1
Feier= c(1,6,81,84,122,133,143,228,306,359,360,361,366
,367,372,466,469,507,518,528,593,671)
Tagestyp[Feier]=3

```

A.2 Zeitreihenanalyse

```

#####
### In richtigen Ordner wechseln
#####

setwd("C:/Users/Tina/Documents/DA/ErgebnisseR")

#####
### Funktionen
#####

periodogramm <- function(m,x) {
T <- length(x)
s <- c(seq( length = (2*T-1), from =(-T+1), by=1 ))
s <- complex(real=0, imag = -s%*%t(m))
  s <- matrix(s,nrow=2*T-1)

gamma(x)%*%exp(s)
}
gamma <- function(x) {

xT <- mean(x)

```

```

T <- length(x)
gamma <- c(1:(2*T-1)) #zum auffüllen und testen

#weise einem vektor die autokovarianzen folgendermaSSen zu:
#fange bei gamma(1) = gamma[-T+1] mit dem auffüllen an ,
#bei T steht dann gamma(0), und ich laufe weiter bis 2*T-1

gamma[T] <- 1/T * sum( (x - xT)*(x - xT) )

for (k in 1:(T-1)){
  xtk <- x[(k+1) : T] #xt-k : erster Faktor der Summanden der k ten Summe
  xt <- x[1 : (T-k)] #zweiter Faktor
  #nutze Symmetrie der Kovarianzen bez 0:

  gamma[T+k] <- 1/T * sum( (xtk - xT)*(xt - xT) )
  gamma[T-k] <- 1/T * sum( (xtk - xT)*(xt - xT) )

}
gamma
}

#####
### Generieren der Matrizen für den Output
#####

mEEX=matrix(0,ncol=27,nrow=19)
vEEX=matrix(0,ncol=27,nrow=19)
acfM=matrix(0,ncol=27,nrow=71) ###Lag plus 1 weil lag Null enthalten ist
pacfM=matrix(0,ncol=27,nrow=71)
acfM2=matrix(0,ncol=27,nrow=366)
pacfM2=matrix(0,ncol=27,nrow=366)
TR=matrix(0,ncol=27,nrow=12)
PerioM=matrix(0,ncol=28,nrow=11) ### Plus 1 um Frequenzen zu haben
ARM=matrix(0,ncol=27,nrow=50)

####Auswahl der Stunde:
for (h in 1:27){

#####
### Mittelwerte und Varianzen
#####

mEEX[1,h]<-mean(EEX[h,1:715])
vEEX[1,h]<-var(EEX[h,1:715])
z=matrix(0,ncol=1,nrow=3)
for (i in 1:715){
  if (Tagestyp[i]==1){
mEEX[2,h]=mEEX[2,h]+EEX[h,i]
vEEX[2,h]=vEEX[2,h]+EEX[h,i]^2
z[1]=z[1]+1

```

```

}
if (Tagestyp[i]==2){
mEEX[3,h]=mEEX[3,h]+EEX[h,i]
vEEX[3,h]=vEEX[3,h]+EEX[h,i]^2
z[2]=z[2]+1
}
if (Tagestyp[i]==3){
mEEX[4,h]=mEEX[4,h]+EEX[h,i]
vEEX[4,h]=vEEX[4,h]+EEX[h,i]^2
z[3]=z[3]+1
}
}
for (i in 2:4){
vEEX[i,h]=1/(z[i-1]-1)*(vEEX[i,h]-1/z[i-1]*mEEX[i,h]^2)
mEEX[i,h]=mEEX[i,h]/z[i-1]
}
for (i in 0:14){
mEEX[i+5,h]=mean(EEX[h,(47*i):(47*(i+1))])
vEEX[i+5,h]=var(EEX[h,(47*i):(47*(i+1))])
}

#####
### ACF und PACF
#####

acfM[,h]<-acf(EEX[h,1:715],lag.max=70,plot=FALSE)$acf
acfM2[,h]<-acf(EEX[h,1:715],lag.max=365,plot=FALSE)$acf
pacfM[,h]<-pacf(EEX[h,1:715],lag.max=71,plot=FALSE)$acf
pacfM2[,h]<-pacf(EEX[h,1:715],lag.max=366,plot=FALSE)$acf

#####
### Trendregression
#####

Trend<-matrix (ncol=4,nrow=715)
Saison<-matrix (ncol=6,nrow=715)
Trend[,1]<-1
Trend[,2]<-seq(715)
Trend[,3]<-seq(715)*seq(715)
Trend[,4]<-seq(715)*seq(715)*seq(715)
Trendregression<-lm(EEX[h,1:715]~Trend[,1]+Trend[,2]+Trend[,3]+Trend[,4])

s<-summary.lm(Trendregression)
TR[c(1,4,7,10),h]<-s$coef[,1]
TR[c(2,5,8,11),h]<-s$coef[,3]
TR[c(3,6,9,12),h]<-s$coef[,4]

#####
### Periodogramm
#####

```

```

#####Tägliche Daten
m<-2*pi/(c(seq(7),14,28,140,365))
Perio <- periodogramm(m,EEX[h,1:715])
Perio<-Re(Perio*Conj(Perio))
PerioM[,1]<-(2*pi)/m
PerioM[, (h+1)]<-m*Re(Perio*Conj(Perio))

#####
### AR Modell Fitten mit AIC Kriterium
#####

AR<-ar(EEX[h,1:715],order.max=25)
ARM[1:(AR$order),h]<-AR$ar
ARM[26:(25+(AR$order)-1),h]<-abs(polyroot(AR$ar))

} ### Ende h-Schleife

#####
### Alte Files löschen In CSV File schreiben
#####

unlink("ErgebnisseMV.csv")
unlink("ErgebnisseCF.csv")
unlink("ErgebnisseTR.csv")
unlink("ErgebnissePerio.csv")
unlink("ErgebnisseAR.csv")
unlink("ErgebnisseCFg.csv")
unlink("ErgebnisseMVg.csv")
unlink("ErgebnisseNSTg.csv")
unlink("ErgebnissePeriog.csv")
unlink("ErgebnisseTRg.csv")
unlink("Ergebnisseg.csv")
unlink("Ergebnisse.csv")

write.csv2(t(EEX),"Ergebnisse.csv",append=TRUE)
write.csv2("mEEX","ErgebnisseMV.csv",append=TRUE)
write.csv2("mEEX","ErgebnisseMV.csv",append=TRUE)
write.csv2("vEEX","ErgebnisseMV.csv",append=TRUE)
write.csv2("vEEX","ErgebnisseMV.csv",append=TRUE)
write.csv2("acfM","ErgebnisseCF.csv",append=TRUE)
write.csv2("acfM","ErgebnisseCF.csv",append=TRUE)
write.csv2("pacfM","ErgebnisseCF.csv",append=TRUE)
write.csv2("pacfM","ErgebnisseCF.csv",append=TRUE)
write.csv2("acfM2","ErgebnisseCF.csv",append=TRUE)
write.csv2("acfM2","ErgebnisseCF.csv",append=TRUE)
write.csv2("pacfM2","ErgebnisseCF.csv",append=TRUE)
write.csv2("pacfM2","ErgebnisseCF.csv",append=TRUE)
write.csv2(TR,"ErgebnisseTR.csv",append=TRUE)
write.csv2(PerioM,"ErgebnissePerio.csv",append=TRUE)

```

```

write.csv2("AR Koefizienten und dazugehörige Nullstellen
des charakteristischen Polynoms", "ErgebnisseAR.csv", append=TRUE)
write.csv2(ARM, "ErgebnisseAR.csv", append=TRUE)

#####
### Gesamte Zeitreihe
#####

#plot(EEXg)
write.csv2(EEXg, "Ergebnisseg.csv", append=TRUE)

#####
### Mittelwerte und Varianzen für gesamte Zeitreihe
#####

mEEXg=matrix(0, ncol=1, nrow=19)
vEEXg=matrix(0, ncol=1, nrow=19)
mEEXg[1]<-mean(EEXg)
vEEXg[1]<-var(EEXg)

z=matrix(0, ncol=1, nrow=3)
for (i in 1:715){
  if (Tagestyp[i]==1){
    mEEXg[2]=mEEXg[2]+sum(EEXg[ ((i-1)*24):(i*24-1)])
    vEEXg[2]=vEEXg[2]+sum(EEXg[ ((i-1)*24):(i*24-1)]^2)
    z[1]=z[1]+1
  }
  if (Tagestyp[i]==2){
    mEEXg[3]=mEEXg[3]+sum(EEXg[ ((i-1)*24):(i*24-1)])
    vEEXg[3]=vEEXg[3]+sum(EEXg[ ((i-1)*24):(i*24-1)]^2)
    z[2]=z[2]+1
  }
  if (Tagestyp[i]==3){
    mEEXg[4]=mEEXg[4]+sum(EEXg[ ((i-1)*24):(i*24-1)])
    vEEXg[4]=vEEXg[4]+EEX[h,i]^2
    z[3]=z[3]+1
  }
}
for (i in 2:4){
  vEEXg[i]=1/(24*z[i-1]-1)*(vEEXg[i]-1/(24*z[i-1])*mEEXg[i]^2)
  mEEXg[i]=mEEXg[i]/(z[i-1]*24)
}

for (i in 0:14){
  mEEXg[i+5]=mean(EEXg[ (47*i*24):(47*24*(i+1))])
  vEEXg[i+5]=var(EEXg[ (47*i*24):(47*24*(i+1))])
}

write.csv2("1. Wert Mittelwert/Varianz,,2. Wert Wochentags,3.Wert Samstags
, 4 Wert Sonntags 5-21 gleitendes Fenster von jeweils 1128 Datenpunkten",

```

```

"ErgebnisseMVg.csv", append=TRUE)
write.csv2(mEEXg, "ErgebnisseMVg.csv", append=TRUE)
write.csv2("mEEXg", "ErgebnisseMVg.csv", append=TRUE)
write.csv2("vEEXg", "ErgebnisseMVg.csv", append=TRUE)
write.csv2("vEEXg", "ErgebnisseMVg.csv", append=TRUE)

#####
### ACF und PACF
#####

a<-acf(EEXg, lag.max=72, plot=FALSE)
write.csv2("acf", "ErgebnisseCFg.csv", append=TRUE)
write.csv2(a$acf, "ErgebnisseCFg.csv", append=TRUE)
a<-pacf(EEXg, lag.max=72, plot=FALSE)
write.csv2("pacf", "ErgebnisseCFg.csv", append=TRUE)
write.csv2(a$acf, "ErgebnisseCFg.csv", append=TRUE)

#####
### Trendregression
#####

Trend<-matrix (ncol=3, nrow=length(EEXg))
Trend[,1]<-1
Trend[,2]<-seq(length(EEXg))/24
Trend[,3]<-seq(length(EEXg))*seq(length(EEXg))/24/24
Trendregression<-lm(EEXg~Trend[,1]+Trend[,2]+Trend[,3]-1)
write.csv2("Ergebnis Trendregression", "ErgebnisseTR.csv", append=TRUE)
s<-summary.lm(Trendregression)
write.csv2(s$coef[,c(1,3,4)], "ErgebnisseTRg.csv", append=TRUE)

#####
### Periodogramm
#####

m<-2*pi/(seq(25))
Perio <- periodogramm(m, EEXg)
Perio<-Re(Perio*Conj(Perio))
write.csv2((2*pi)/m, "ErgebnissePeriog.csv", append=TRUE)
write.csv2(m*(Perio), "ErgebnissePeriog.csv", append=TRUE)

```

A.3 Modelle bauen

```

#####
### In richtigen Ordner wechseln
#####

setwd("C:/Users/Tina/Documents/DA/Programme")

#####

```

```

### Basis Modell für Stunde h nur alte Preise
#####
ModellKlein<-function (Tagestyp,EEX,h,D,M,a) {
#a gibt an ob mit (2) oder ohne (1) zusätzlichen Parametern
#M Anzahl der Parameter
X<-matrix(ncol=M,nrow=D)
Xtemp<-matrix(ncol=3,nrow=D)
Y<-EEX[h,]

#Dummie für Sa&So
for (i in 1:D){
if (Tagestyp[i]==2){
X[i,1]=1
}
if (Tagestyp[i]==3){
X[i,1]=1
}
if (Tagestyp[i]==1){
X[i,1]=0
}
}
#Dummie für So
for (i in 1:D){
if (Tagestyp[i]==2){
X[i,2]=0
}
if (Tagestyp[i]==3){
X[i,2]=1
}
if (Tagestyp[i]==1){
X[i,2]=0
}
}
#Durchschnitt letzten 5 Wertktage
Xtemp[,1]=0
for (i in 9:D){
j=1
k=1
while (j<6){
if (X[i-k,1]==1){
k=k+1
}
if (X[i-k,1]==0){
Xtemp[i,1]=Xtemp[i,1]+Y[i-k]
j=j+1
k=k+1
}
}
}
Xtemp[,1]=Xtemp[,1]/5

```

```

#Durchschnitt letzten 5 Samstage
Xtemp[,2]=0
for (i in 36:D){
j=1
k=1
while (j<6){
if (Tagestyp[i-k]==1){
k=k+1
}
if (Tagestyp[i-k]==3){
k=k+1
}
if (Tagestyp[i-k]==2){
Xtemp[i,2]=Xtemp[i,2]+Y[i-k]
j=j+1
k=k+1
}
}
}
Xtemp[,2]=Xtemp[,2]/5
#Durchschnitt letzten 5 Sonntage
Xtemp[,3]=0
for (i in 36:D){
j=1
k=1
while (j<6){
if (Tagestyp[i-k]==1){
k=k+1
}
if (Tagestyp[i-k]==2){
k=k+1
}
if (Tagestyp[i-k]==3){
Xtemp[i,3]=Xtemp[i,3]+Y[i-k]
j=j+1
k=k+1
}
}
}
Xtemp[,3]=Xtemp[,3]/5
X[,3]=0
for (i in 1:D){
if (Tagestyp[i]==1){
X[i,3]=Xtemp[i,1]
}
if (Tagestyp[i]==2){
X[i,3]=Xtemp[i,2]
}
if (Tagestyp[i]==3){
X[i,3]=Xtemp[i,3]
}
}

```

```

}
}
#Letzter Werktag
Xtemp[,1]=0
for (i in 7:D){
j=1
k=1
while (j<2){
if (X[i-k,1]==1){
k=k+1
}
if (X[i-k,1]==0){
Xtemp[i,1]=Y[i-k]
j=j+1
k=k+1
}
}
}
#Letzter Samstag
Xtemp[,2]=0
for (i in 7:D){
j=1
k=1
while (j<2){
if (Tagestyp[i-k]==1){
k=k+1
}
if (Tagestyp[i-k]==3){
k=k+1
}
if (Tagestyp[i-k]==2){
Xtemp[i,2]=Y[i-k]
j=j+1
k=k+1
}
}
}
#Letzter Sonntag
Xtemp[,3]=0
for (i in 7:D){
j=1
k=1
while (j<2){
if (Tagestyp[i-k]==1){
k=k+1
}
if (Tagestyp[i-k]==2){
k=k+1
}
}
}

```

```

if (Tagestyp[i-k]==3){
Xtemp[i,3]=Y[i-k]
j=j+1
k=k+1
}
}
}
X[,4]=0
for (i in 1:D){
if (Tagestyp[i]==1){
X[i,4]=Xtemp[i,1]
}
if (Tagestyp[i]==2){
X[i,4]=Xtemp[i,2]
}
if (Tagestyp[i]==3){
X[i,4]=Xtemp[i,3]
}
}
}

#Durchschnitt der letzten 7 Tage
X[,5]=0
for (i in 8:D){
j=1
k=1
while (j<8){
X[i,5]=X[i,5]+Y[i-k]
j=j+1
k=k+1
}
}
X[,5]=X[,5]/7

if (a==2){
X[,6]<-Wind[h,1:D]
X[,7]<-Prognose[7,1:D]
X[,8]<-Prognose[14,1:D]
X[,9]<-EXAA[h,1:D]
}
return(X)
}

#####
### GroSSes Modell für Stunde h
#####

ModellGrosz<-function (Tagestyp,EEX,h,D,M,a){
X=ModellKlein(Tagestyp,EEX,h,D,M,a)
k=0

```

```

while (is.na(X[1, (k+1)])==FALSE){
k=k+1
}
m=k+1
for (i in 1:k){
for (j in 1:k){
if(j>=i && j<=k){
if (sum(X[1:D,i]*X[1:D,j]-X[1:D,j])!=0){
X[,m]=X[,i]*X[,j]
m=m+1
#print(m-1)      ### Index von Theta
#print(c(i,j))   ###Zum Modell nachvollziehen
}
}
}
}

return(X)
}

```

A.4 AIC Algorithmus

```

#####
### In richtigen Ordner wechseln
#####

setwd("C:/Users/Tina/Documents/DA/Programme")

#####
### AIC Kriterium
#####

aicforward<-function (Data){ ##### Daten müssen Spaltenweise sein
-> Anzahl der Spalten entspricht Anzahl der Regressoren
l=length(Data[,1])
aic=matrix(ncol=(l+1),nrow=1,0)   ### Forward Algorithm Teil 1
immer eine Spalte dazu (diejenige mit bestem AIC)
for (i in 1:l){
for (j in 1:l){
if (aic[i, (j+1)]==0){
aic[i, (j+1)]=j
data=as.data.frame(Data[1:D,c(aic[i,2:(l+1)])])
reg<-lm(Y~.,data)
min=AIC(reg)
if (aic[i,1]==0){
aic[i,1]=min
}
if (min<=aic[i,1]){
aic[i,1]=min
}
}
}
}
}

```

```

k=j
}
aic[i, (j+1)]=0
}
}
aic[i:l, (k+1)]=k
}
pos=which.min(aic[,1])
aic2=matrix(ncol=(l+1),nrow=1,0)
aic2=aic[pos,]
return (aic2)
}

aicsecondstep<-function (Data,aicin){
l=length(Data[1,])
aicnew=matrix(ncol=(l+1),nrow=1,0)
ende=0
count=0
while (ende<aicin[1]){
count=count+1
for (i in 1:l){
if (aicin[(i+1)]==0){
aicnew[i,]=aicin
aicnew[i,i+1]=i
reg<-lm(Y~.,data=as.data.frame(Data[1:D,c(aicnew[i,2:(l+1)])]))
min=AIC(reg)
aicnew[i,1]=min
}
if (aicin[(i+1)]!=0 && (sum(aicin[2:(l+1)])-aicin[(i+1)])!=0){
aicnew[i,]=aicin
aicnew[i,i+1]=0
reg<-lm(Y~.,data=as.data.frame(Data[1:D,c(aicnew[i,2:(l+1)])]))
min=AIC(reg)
aicnew[i,1]=min
}
}
pos=which.min(aicnew[,1])
if (aicin[1]<=aicnew[pos,1]){
ende=aicin[1]
}
if (aicin[1]>aicnew[pos,1]){
aicin=aicnew[pos,]
}
}
return(aicin)
}

```

A.5 Naive Prognose

```
#####  
### Modell für Stunde h  
#####  
  
D=711  
P=364      ### Länge des Intervals zum kalibrieren  
I=7        ### Länge des verschiebungsintervalls  
  
MSE=matrix(0,nrow=4,ncol=27)  
MAE=matrix(0,nrow=4,ncol=27)  
Hitrate=matrix(0,nrow=1,ncol=27)  
  
for (h in 1:27){  
  Y<-EEX[h,1:D]  
  x=1  
  
#####  
### Generieren der Modelle  
#####  
  
X<-ModellKlein(Tagestyp,EEX,h,D,5,1)  
X2<-ModellKlein(Tagestyp,EEX,h,D,9,2)  
  
#####  
### Naive Prognose  
#####  
  
P=364      ### Länge des Intervals zum kalibrieren  
I=7        ### Länge des verschiebungsintervalls  
preN=matrix(0,ncol=1,nrow=D)  
  
for (i in 2:D){  
  preN[i]=X[i,4]  
  }  
  resN<-preN[(P+1):D]-Y[(P+1):D]  
  resNw=0  
  resNSASO=0  
  resNSO=0  
  for (i in 1:length(resN)){  
    if (Tagestyp[364+i]==1){  
      resNw=c(resNw,resN[i])  
    }  
    if (Tagestyp[364+i]==2){  
      resNSASO=c(resNSASO,resN[i])  
    }  
    if (Tagestyp[364+i]==3){  
      resNSO=c(resNSO,resN[i])  
    }  
  }  
}
```

```

}

###Mittlerer Quadratischer Fehler
MSE[1,h]=sum(resN^2)*(1/length(resN))
MSE[2,h]=sum(resNw^2)*(1/length(resNw))
MSE[3,h]=sum(resNSASO^2)*(1/length(resNSASO))
MSE[4,h]=sum(resNSO^2)*(1/length(resNSO))

###Mittlerer Absoluter Fehler
MAE[1,h]=sum(abs(resN))*(1/length(resN))
MAE[2,h]=sum(abs(resNw))*(1/length(resNw))
MAE[3,h]=sum(abs(resNSASO))*(1/length(resNSASO))
MAE[4,h]=sum(abs(resNSO))*(1/length(resNSO))

###Hitrate
for (i in (P+1):(D-4)){
  if (preN[i-P]-Y[i-1]>0 && Y[i]-Y[i-1]>0 ||
  preN[i-P]-Y[i-1]<0 && Y[i]-Y[i-1]<0){
    Hitrate[1,h]=Hitrate[1,h]+1
  }
}

print(h)

}
setwd("C:/Users/Tina/Documents/DA/ErgebnisseR")
unlink("PrognoseNaive.csv")      ### Altes File löschen
write.csv2("MAE","PrognoseNaive.csv",append=TRUE)
write.csv2("MAE","PrognoseNaive.csv",append=TRUE)
write.csv2("MSE","PrognoseNaive.csv",append=TRUE)
write.csv2("MSE","PrognoseNaive.csv",append=TRUE)
Hitrate=Hitrate/((D-4)-(P+1))
write.csv2("Hitrate","PrognoseNaive.csv",append=TRUE)
write.csv2(Hitrate,"PrognoseNaive.csv",append=TRUE)

%

```

A.6 Kleinste Quadrate Prognose

```

%

#####
### Modell für Stunde h
#####

D=711
P=364      ### Länge des Intervalls zum kalibrieren
I=7        ### Länge des verschiebungsintervalls

MSE=matrix(0,nrow=16,ncol=27)

```

```

MAE=matrix(0,nrow=16,ncol=27)
Hitrate=matrix(0,nrow=4,ncol=27)
VEC=matrix(0,nrow=(50*4),ncol=(27*2))
a=matrix(0,nrow=26,ncol=4)

for (h in 1:27){
for (x in 1:2){
Y<-EEX[h,1:D]

#####
### Generieren der Modelle
#####

if (x == 1){
X<-ModellKlein(Tagestyp,EEX,h,D,5,1)
X2<-ModellKlein(Tagestyp,EEX,h,D,9,2)
}
if (x == 2){
X<-ModellGrosz(Tagestyp,EEX,h,D,17,1)
X2<-ModellGrosz(Tagestyp,EEX,h,D,51,2)
}

#####
### OLS nur alte Preise
#####

aic1<-aicforward(as.data.frame(X[1:365,]))
preOLSP=matrix(ncol=1,nrow=(D-P-4))
for (i in 0:((D-P)%/I-1)){
aic1<-aicsecondstep(as.data.frame(X[(1+I*i):(P+I*i),]),aic1)
vec1<-c(aic1[2:length(aic1)])
VEC[(1+(50)*2*(x-1)):(length(vec1)+(50)*2*(x-1)),h]=
VEC[(1+(50)*2*(x-1)):(length(vec1)+(50)*2*(x-1)),h]+vec1
datareg=as.data.frame(X[(1+I*i):(P+I*i),vec1])
reg<-lm(Y[(1+I*i):(P+I*i)]~.,datareg)
preOLSP[(i*I+1):((i+1)*I)]<-
predict(reg,as.data.frame(X[(P+i*I+1):(P+(i+1)*I),vec1]))
}
resOLSP<-preOLSP-Y[(P+1):(D-4)]

resOLSPw=0
resOLSPSASO=0
resOLSPSO=0
for (i in 1:length(resOLSP)){
if (Tagestyp[364+i]==1){
resOLSPw=c(resOLSPw,resOLSP[i])
}
if (Tagestyp[364+i]==2){
resOLSPSASO=c(resOLSPSASO,resOLSP[i])
}
}

```

```

if (Tagestyp[364+i]==3){
resOLSPSO=c(resOLSPSO,resOLSP[i])
}
}

for (i in 1:length(resOLSP)){
if (is.na(resOLSP[i])==TRUE){
resOLSP[i]=0
}
}
b<-acf(resOLSP)
a[, (2*(x-1)+1)]<-b$acf

#####
### OLS
#####

aic2<-aicforward(as.data.frame(X2[1:365,]))
preOLS=matrix(ncol=1,nrow=(D-P-4))
for (i in 0:((D-P)%/I-1)){
Data=as.data.frame(X2[(1+I*i):(P+I*i),])
aic2<-aicsecondstep(Data,aic2)
vec2<-c(aic2[2:length(aic2)])
VEC[((49)+(50)*2*(x-1)+1):((49)+length(vec2)+(50)*2*(x-1)),h]=
VEC[((49)+(50)*2*(x-1)+1):((49)+length(vec2)+(50)*2*(x-1)),h]+vec2
datareg=as.data.frame(X2[(1+I*i):(P+I*i),vec2])
reg<-lm(Y[(1+I*i):(P+I*i)]~.,datareg)
preOLS[(i*I+1):((i+1)*I)]<-
predict(reg,as.data.frame(X2[(P+i*I+1):(P+(i+1)*I),vec2]))
}
resOLS<-preOLS-Y[(P+1):(D-4)]

resOLSw=0
resOLSSASO=0
resOLSSO=0
for (i in 1:length(resOLS)){
if (Tagestyp[364+i]==1){
resOLSw=c(resOLSw,resOLS[i])
}
if (Tagestyp[364+i]==2){
resOLSSASO=c(resOLSSASO,resOLS[i])
}
if (Tagestyp[364+i]==3){
resOLSSO=c(resOLSSO,resOLS[i])
}
}
for (i in 1:length(resOLS)){
if (is.na(resOLS[i])==TRUE){
resOLS[i]=0
}
}

```

```

}

b<-acf(resOLS)
a[, (2*(x-1)+2)]<-b$acf

#####
### Prognosefehler
#####

###Mittlerer Quadratischer Fehler
MSE[(8*(x-1)+1),h]=sum(resOLSP^2)*(1/length(resOLSP))
MSE[(8*(x-1)+2),h]=sum(resOLS^2)*(1/length(resOLS))
MSE[(8*(x-1)+3),h]=sum(resOLSPw^2)*(1/length(resOLSPw))
MSE[(8*(x-1)+4),h]=sum(resOLSw^2)*(1/length(resOLSw))
MSE[(8*(x-1)+5),h]=sum(resOLSPSASO^2)*(1/length(resOLSPSASO))
MSE[(8*(x-1)+6),h]=sum(resOLSSASO^2)*(1/length(resOLSSASO))
MSE[(8*(x-1)+7),h]=sum(resOLSPSO^2)*(1/length(resOLSPSO))
MSE[(8*(x-1)+8),h]=sum(resOLSSO^2)*(1/length(resOLSSO))

###Mittlerer Absoluter Fehler
MAE[(8*(x-1)+1),h]=sum(abs(resOLSP))*(1/length(resOLSP))
MAE[(8*(x-1)+2),h]=sum(abs(resOLS))*(1/length(resOLS))
MAE[(8*(x-1)+3),h]=sum(abs(resOLSPw))*(1/length(resOLSPw))
MAE[(8*(x-1)+4),h]=sum(abs(resOLSw))*(1/length(resOLSw))
MAE[(8*(x-1)+5),h]=sum(abs(resOLSPSASO))*(1/length(resOLSPSASO))
MAE[(8*(x-1)+6),h]=sum(abs(resOLSSASO))*(1/length(resOLSSASO))
MAE[(8*(x-1)+7),h]=sum(abs(resOLSPSO))*(1/length(resOLSPSO))
MAE[(8*(x-1)+8),h]=sum(abs(resOLSSO))*(1/length(resOLSSO))

###Hitrate
for (i in (P+1):(D-4)){

if (preOLSP[i-P]-Y[i-1]>0 && Y[i]-Y[i-1]>0 ||
preOLSP[i-P]-Y[i-1]<0 && Y[i]-Y[i-1]<0){
Hitrate[(2*(x-1)+1),h]=Hitrate[(2*(x-1)+1),h]+1

}

if (preOLS[i-P]-Y[i-1]>0 && Y[i]-Y[i-1]>0 || pre
OLS[i-P]-Y[i-1]<0 && Y[i]-Y[i-1]<0){
Hitrate[(2*(x-1)+2),h]=Hitrate[(2*(x-1)+2),h]+1

}

}

print(h)

}
}
###Für h Schleife

```

```

setwd("C:/Users/Tina/Documents/DA/ErgebnisseR")
unlink("PrognoseOLS.csv")      ### Altes File löschen
write.csv2("MAE", "PrognoseOLS.csv", append=TRUE)
write.csv2(MAE, "PrognoseOLS.csv", append=TRUE)
write.csv2("MSE", "PrognoseOLS.csv", append=TRUE)
write.csv2(MSE, "PrognoseOLS.csv", append=TRUE)
Hitrate=Hitrate/((D-4)-(P+1))
write.csv2("Hitrate", "PrognoseOLS.csv", append=TRUE)
write.csv2(Hitrate, "PrognoseOLS.csv", append=TRUE)
write.csv2("ACF", "PrognoseOLS.csv", append=TRUE)
write.csv2(a, "PrognoseOLS.csv", append=TRUE)
write.csv2(VEC, "PrognoseOLS.csv", append=TRUE)

```

⊘

A.7 Verallgemeinerte kleinste Quadrate Prognose

⊘

```

#####
###Modell für Stunde h
#####

D=711
P=364      ### Länge des Intervalls zum kalibrieren
I=7        ### Länge des verschiebungsintervalls

MSE=matrix(0,nrow=16,ncol=27)
MAE=matrix(0,nrow=16,ncol=27)
Hitrate=matrix(0,nrow=4,ncol=27)
VEC=matrix(0,nrow=(50*4),ncol=(27*2))
a=matrix(0,nrow=26,ncol=4)

EEXm=EEX

for (h in 1:27){
  for (x in 1:2){
    Y<-EEX[h,1:D]

#####
### Generieren der Modelle
#####

    if (x == 1){
      X<-ModellKlein(Tagestyp,EEX,h,D,5,1)
      X2<-ModellKlein(Tagestyp,EEX,h,D,9,2)
    }
    if (x == 2){
      X<-ModellGrosz(Tagestyp,EEX,h,D,17,1)
      X2<-ModellGrosz(Tagestyp,EEX,h,D,51,2)
    }
  }
}

```

```

}

#####
### Gewichte generieren
#####

aic1<-aicforward(as.data.frame(X[1:365,]))
aictemp<-aicsecondstep(as.data.frame(X[1:365,]),aic1)
datareg=as.data.frame(X[1:365,],aictemp)
reg<-lm(Y[1:365]~.,datareg)
res=reg$residuals
Tagesdummies=matrix(0,nrow=365,ncol=3)
Tagesdummies[,1]=1-X[1:365,1]
Tagesdummies[,2]=X[1:365,1]-X[1:365,2]
Tagesdummies[,3]=X[1:365,2]
regres<-lm(res^2~Tagesdummies[,1]+Tagesdummies[,2]+Tagesdummies[,3]-1)
regres$coefficients=regres$coefficients^(1/2)
for (i in 1:length(X[1,])){
for (j
in 1: D){
if (Tagestyp[j]==1 && X[j,i]!=0){
X[j,i]=X[j,i]/(regres$coefficients[1])
Ym[j]=EEX[h,j]/(regres$coefficients[1])
}
if
(Tagestyp[j]==2 && X[j,i]!=0){
X[j,i]=X[j,i]/(regres$coefficients[2])
Ym[j]=EEX[h,j]/(regres$coefficients[2])
}
if
(Tagestyp[j]==3 && X[j,i]!=0){
X[j,i]=X[j,i]/(regres$coefficients[3])
Ym[j]=EEX[h,j]/(regres$coefficients[3])
}
}}

#####
### OLS nur alte Preise
#####

aic1<-aicforward(as.data.frame(X[1:365,]))
preOLSP=matrix(ncol=1,nrow=(D-P-4))
for (i in 0:((D-P)%/I-1)){
aic1<-aicsecondstep(as.data.frame(X[(1+I*i):(P+I*i)],),aic1)
vec1<-c(aic1[2:length(aic1)])
VEC[(1+(50)*2*(x-1)):(length(vec1)+(50)*2*(x-1)),h]=
VEC[(1+(50)*2*(x-1)):(length(vec1)+(50)*2*(x-1)),h]+vec1
datareg=as.data.frame(X[(1+I*i):(P+I*i)],vec1)
reg<-lm(Ym[(1+I*i):(P+I*i)]~.,datareg)

```

```

preOLSP[(i*I+1):(i+1)*I]<-
predict(reg,as.data.frame(X[(P+i*I+1):(P+(i+1)*I),vec1]))
}

for (i in 1: length(preOLSP)){
if (Tagestyp[364+i]==1){
preOLSP[i]=preOLSP[i]*regres$coefficients[1]
}
if
(Tagestyp[364+i]==2){
preOLSP[i]=preOLSP[i]*regres$coefficients[2]
}
if
(Tagestyp[364+i]==3){
preOLSP[i]=preOLSP[i]*regres$coefficients[3]
}
}
resOLSP<-preOLSP-Y[(P+1):(D-4)]

resOLSPw=0
resOLSPSASO=0
resOLSPSO=0
for (i in 1: length(resOLSP)){
if (Tagestyp[364+i]==1){
resOLSPw=c(resOLSPw,resOLSP[i])
}
if (Tagestyp[364+i]==2){
resOLSPSASO=c(resOLSPSASO,resOLSP[i])
}
if (Tagestyp[364+i]==3){
resOLSPSO=c(resOLSPSO,resOLSP[i])
}
}

for (i in 1:length(resOLSP)){
if (is.na(resOLSP[i])==TRUE){
resOLSP[i]=0
}
}

resOLSP2=resOLSP
for (i in 1: length(resOLSP)){ ###Für ACF
if (Tagestyp[364+i]==1){
resOLSP2[i]=resOLSP[i]/(regres$coefficients[1])
}
if
(Tagestyp[364+i]==2){
resOLSP2[i]=resOLSP[i]/(regres$coefficients[2])
}
if

```

```

(Tagestyp[364+i]==3) {
resOLSP2[i]=resOLSP[i]/(regres$coefficients[3])
}
}
b<-acf(resOLSP2)
a[, (2*(x-1)+1)]<-b$acf

#####
### Gewichte
#####

Data=as.data.frame(X2[1:365,])
aic2<-aicforward(as.data.frame(X2[1:365,]))
aictemp<-aicsecondstep(Data,aic2)
datareg=as.data.frame(X2[1:365,],aictemp)
reg<-lm(Y[1:365]~.,datareg)
res=reg$residuals
regres<-lm(res^2~Tagesdummies[,1]+Tagesdummies[,2]+Tagesdummies[,3]-1)
regres$coefficients=regres$coefficients^(1/2)
for (i in 1:length(X2[1,])){
for
(j in 1: D){
if (Tagestyp[j]==1 && X2[j,i]!=0){
X2[j,i]=X2[j,i]/(regres$coefficients[1])
Ym[j]=EEX[h,j]/(regres$coefficients[1])
}
if
(Tagestyp[j]==2 && X2[j,i]!=0){
X2[j,i]=X2[j,i]/(regres$coefficients[2])
Ym[j]=EEX[h,j]/(regres$coefficients[2])
}
if
(Tagestyp[j]==3 && X2[j,i]!=0){
X2[j,i]=X2[j,i]/(regres$coefficients[3])
Ym[j]=EEX[h,j]/(regres$coefficients[3])
}
}}

#####
### OLS
#####

aic2<-aicforward(as.data.frame(X2[1:365,]))
preOLS=matrix(ncol=1,nrow=(D-P-4))
for (i in 0:((D-P)%/I-1)){
Data=as.data.frame(X2[(1+I*i):(P+I*i),])
aic2<-aicsecondstep(Data,aic2)
vec2<-c(aic2[2:length(aic2)])
VEC[((49)+(50)*2*(x-1)+1):((49)+length(vec2)+(50)*2*(x-1)),h]+vec2

```

```

datareg=as.data.frame(X2[(1+I*i):(P+I*i),vec2])
reg<-lm(Ym[(1+I*i):(P+I*i)]~.,datareg)
preOLS[(i*I+1):((i+1)*I)]<-
predict(reg,as.data.frame(X2[(P+i*I+1):(P+(i+1)*I),vec2]))
}

for (i in 1: length(preOLS[])){
if (Tagestyp[364+i]==1){
preOLS[i]=preOLS[i]*regres$coefficients[1]
}
if (Tagestyp[364+i]==2){
preOLS[i]=preOLS[i]*regres$coefficients[2]
}
if (Tagestyp[364+i]==3){
preOLS[i]=preOLS[i]*regres$coefficients[3]
}
}
resOLS<-preOLS-Y[(P+1):(D-4)]

resOLSw=0
resOLSSASO=0
resOLSSO=0
for (i in 1: length(resOLS)){
if (Tagestyp[364+i]==1){
resOLSw=c(resOLSw,resOLS[i])
}
if (Tagestyp[364+i]==2){
resOLSSASO=c(resOLSSASO,resOLS[i])
}
if (Tagestyp[364+i]==3){
resOLSSO=c(resOLSSO,resOLS[i])
}
}

resOLS2=resOLS
for (i in 1: length(resOLS)){ ###Für ACF
if (Tagestyp[364+i]==1){
resOLS2[i]=resOLS[i]/(regres$coefficients[1])
}
if
(Tagestyp[364+i]==2){
resOLS2[i]=resOLS[i]/(regres$coefficients[2])
}
if
(Tagestyp[364+i]==3){
resOLS2[i]=resOLS[i]/(regres$coefficients[3])
}
}
}
b<-acf(resOLS2)
a[, (2*(x-1)+2)]<-b$acf

```

```

#####
### Prognosefehler
#####

###Mittlerer Quadratischer Fehler
MSE[(8*(x-1)+1),h]=sum(resOLSP^2)*(1/length(resOLSP))
MSE[(8*(x-1)+2),h]=sum(resOLS^2)*(1/length(resOLS))
MSE[(8*(x-1)+3),h]=sum(resOLSPw^2)*(1/length(resOLSPw))
MSE[(8*(x-1)+4),h]=sum(resOLSw^2)*(1/length(resOLSw))
MSE[(8*(x-1)+5),h]=sum(resOLSPSASO^2)*(1/length(resOLSPSASO))
MSE[(8*(x-1)+6),h]=sum(resOLSSASO^2)*(1/length(resOLSSASO))
MSE[(8*(x-1)+7),h]=sum(resOLSPSO^2)*(1/length(resOLSPSO))
MSE[(8*(x-1)+8),h]=sum(resOLSSO^2)*(1/length(resOLSSO))

###Mittlerer Absoluter Fehler
MAE[(8*(x-1)+1),h]=sum(abs(resOLSP))*(1/length(resOLSP))
MAE[(8*(x-1)+2),h]=sum(abs(resOLS))*(1/length(resOLS))
MAE[(8*(x-1)+3),h]=sum(abs(resOLSPw))*(1/length(resOLSPw))
MAE[(8*(x-1)+4),h]=sum(abs(resOLSw))*(1/length(resOLSw))
MAE[(8*(x-1)+5),h]=sum(abs(resOLSPSASO))*(1/length(resOLSPSASO))
MAE[(8*(x-1)+6),h]=sum(abs(resOLSSASO))*(1/length(resOLSSASO))
MAE[(8*(x-1)+7),h]=sum(abs(resOLSPSO))*(1/length(resOLSPSO))
MAE[(8*(x-1)+8),h]=sum(abs(resOLSSO))*(1/length(resOLSSO))

###Hitrate
for (i in (P+1):(D-4)){

if (preOLSP[i-P]-Y[i-1]>0 && Y[i]-Y[i-1]>0 || pre
OLSP[i-P]-Y[i-1]<0 && Y[i]-Y[i-1]<0){
Hitrate[(2*(x-1)+1),h]=Hitrate[(2*(x-1)+1),h]+1

}
if (preOLS[i-P]-Y[i-1]>0 && Y[i]-Y[i-1]>0 || pre
OLS[i-P]-Y[i-1]<0 && Y[i]-Y[i-1]<0){
Hitrate[(2*(x-1)+2),h]=Hitrate[(2*(x-1)+2),h]+1

}
}

print(h)

}
}
###Für h Schleife

setwd("C:/Users/Tina/Documents/DA/ErgebnisseR")
unlink("PrognoseOLSG.csv")          ### Altes File löschen
write.csv2("MAE", "PrognoseOLSG.csv", append=TRUE)
write.csv2(MAE, "PrognoseOLSG.csv", append=TRUE)

```

```

write.csv2("MSE","PrognoseOLSG.csv",append=TRUE)
write.csv2(MSE,"PrognoseOLSG.csv",append=TRUE)
Hitrate=Hitrate/((D-4)-(P+1))
write.csv2("Hitrate","PrognoseOLSG.csv",append=TRUE)
write.csv2(Hitrate,"PrognoseOLSG.csv",append=TRUE)
write.csv2("ACF","PrognoseOLSG.csv",append=TRUE)
write.csv2(a,"PrognoseOLSG.csv",append=TRUE)
write.csv2(VEC,"PrognoseOLSG.csv",append=TRUE)

```

⌘

A.8 Support Vector Regression mit Gewichten

⌘

```

#####
###Bibliotheken laden und in richtigen Ordner wechseln
#####

library(e1071)
library(class)
library(zoo)
library(lmtest)
library(nlme)
library(kernlab)
setwd("C:/Users/Tina/Documents/DA/Programme")

kernelsvm=c("linear","polynomial","radial","sigmoid")
typesvm=c("eps-regression","nu-regression")

#####
### Modell für Stunde h
#####

D=711
P=364      ### Länge des Intervalls zum Kalibrieren
I=7        ### Länge des verschiebungsintervalls

MSE=matrix(0,nrow=16,ncol=27)
MAE=matrix(0,nrow=16,ncol=27)
Hitrate=matrix(0,nrow=16,ncol=27)

for (h in 1:27){
Y<-EEX[h,1:D]

#####
### Generieren der Modelle
#####

```

```

if (x == 1){
X<-ModellKlein(Tagestyp,EEX,h,D,5,1)
X2<-ModellKlein(Tagestyp,EEX,h,D,9,2)
}
if (x == 2){
X<-ModellGrosz(Tagestyp,EEX,h,D,17,1)
X2<-ModellGrosz(Tagestyp,EEX,h,D,51,2)
}

#####
### SVM nur alte Preise
#####

pos=0
for (n in 1: length(typesvm)){
pos=0
for (k in 1: length(kernelsvm)){
pos=pos+1
preSVRP=matrix(0,ncol=(length(typesvm)*length(kernelsvm)),nrow=(D-P))
resSVRP=matrix(0,ncol=(length(typesvm)*length(kernelsvm)),nrow=(D-P))
for (i in 0:((D-P)%/I-1)){
svr<-svm(x=X[(1+I*i):(P+I*i)],y=Y[(1+I*i):(P+I*i)],
kernel=kernelsvm[k],type=typesvm[n],coef=0.25)
preSVRP[(i*I+1):((i+1)*I),pos]<-predict(svr,X[(P+i*I+1):(P+(i+1)*I),])
}

resSVRP[,pos]<-preSVRP[1:(D-P),pos]-Y[(P+1):D]

#####
### SVM
#####

preSVR=matrix(0,ncol=(length(typesvm)*length(kernelsvm)),nrow=(D-P))
resSVR=matrix(0,ncol=(length(typesvm)*length(kernelsvm)),nrow=(D-P))
for (i in 0:((D-P)%/I-2)){
svr<-svm(x=X2[(1+I*i):(P+I*i)],y=Y[(1+I*i):(P+I*i)],
kernel=kernelsvm[k],type=typesvm[n],coef=0.25)
tune.svm(x=X2[(1+I*i):(P+I*i)],y=Y[(1+I*i):(P+I*i)])
preSVR[(i*I+1):((i+1)*I),pos]<-predict(svr,X2[(P+i*I+1):(P+(i+1)*I),])
}
resSVR[,pos]<-preSVR[,pos]-Y[(P+1):D]

#####
### Prognosefehler
#####

if (n==1){
c=0
}
if (n==2){

```

```

c=4
}
###Mittlerer Quadratischer Fehler
MSE[(pos+c),h]=sum(resSVRP[,pos]^2)*(1/length(resSVRP[,pos]))
MSE[(pos+8+c),h]=sum(resSVR[,pos]^2)*(1/length(resSVR[,pos]))
###Mittlerer Absoluter Fehler
MAE[(pos+c),h]=sum(abs(resSVRP[,pos]))*(1/length(resSVRP[,pos]))
MAE[(pos+8+c),h]=sum(abs(resSVR[,pos]))*(1/length(resSVR[,pos]))
###Hitrate
for (i in (P+1):(D-4)){
if (preSVRP[i-P,pos]-Y[i-1]>0 && Y[i]-Y[i-1]>0 || pre
SVRP[i-P,pos]-Y[i-1]<0 && Y[i]-Y[i-1]<0){
Hitrate[(pos+c),h]=Hitrate[(pos+c),h]+1
}
if (preSVR[i-P,pos]-Y[i-1]>0 && Y[i]-Y[i-1]>0 || pre
SVR[i-P,pos]-Y[i-1]<0 && Y[i]-Y[i-1]<0){
Hitrate[(pos+8+c),h]=Hitrate[(pos+c+8),h]+1
}
}

print(h)
}
}

}###Für h Schleife

setwd("C:/Users/Tina/Documents/DA/ErgebnisseR")
unlink("PrognoseSVR.csv")      ### Altes File löschen
write.csv2(kernelsvm,"PrognoseSVR.csv",append=TRUE)
write.csv2(typesvm,"PrognoseSVR.csv",append=TRUE)
write.csv2("MAE","PrognoseSVR.csv",append=TRUE)
write.csv2("MAE","PrognoseSVR.csv",append=TRUE)
write.csv2("MSE","PrognoseSVR.csv",append=TRUE)
write.csv2("MSE","PrognoseSVR.csv",append=TRUE)
Hitrate=Hitrate/((D-4)-(P+1))
write.csv2("Hitrate","PrognoseSVR.csv",append=TRUE)
write.csv2(Hitrate,"PrognoseSVR.csv",append=TRUE)

```

%

A.9 Support Vector Regression

%

```

#####
### Bibliotheken laden und in richtigen Ordner wechseln
#####

library(e1071)

```

```

library(class)
library(zoo)
library(lmtest)
library(nlme)
library(kernlab)
setwd("C:/Users/Tina/Documents/DA/Programme")
kernelsvm=c("linear","polynomial","radial","sigmoid")
typesvm=c("eps-regression","nu-regression")

#####
### Modell für Stunde h
#####

D=711
P=364      ### Länge des Intervalls zum kalibrieren
I=7        ### Länge des verschiebungsintervalls

MSE=matrix(0,nrow=16,ncol=27)
MAE=matrix(0,nrow=16,ncol=27)
Hitrate=matrix(0,nrow=16,ncol=27)
EEXm=EEX

for (h in 1:27){
Y<-EEX[h,1:D]
Ym=Y
pos=0
for (n in 1: length(typesvm)){
pos=0
for (k in 1: length(kernelsvm)){
pos=pos+1

#####
### Generieren der Modelle
#####

x=1
if (x == 1){
X<-ModellKlein(Tagestyp,EEX,h,D,5,1)
X2<-ModellKlein(Tagestyp,EEX,h,D,9,2)
}
if (x == 2){
X<-ModellGrosz(Tagestyp,EEX,h,D,17,1)
X2<-ModellGrosz(Tagestyp,EEX,h,D,51,2)
}
#####
### Gewichte EEXm generieren
#####

datareg=as.data.frame(X[1:365,])

```

```

reg<-lm(Y[1:365]~.,datareg)
svr<-svm(x=X[1:365,],y=Y[1:365],kernel=kernelsvm[k],type=typesvm[n],coef=0.25)
res=svr$fitted-Y[1:365]

Tagesdummies=matrix(0,nrow=365,ncol=3)
Tagesdummies[,1]=1-X[1:365,1]
Tagesdummies[,2]=X[1:365,1]-X[1:365,2]
Tagesdummies[,3]=X[1:365,2]
res2=res^2
regres<-lm(res2~Tagesdummies[,1]+Tagesdummies[,2]+Tagesdummies[,3]-1)
regres$coefficients=regres$coefficients^(1/2)
for (i in 1:5){
for (j in 1:D){
if (Tagestyp[j]==1 && X[j,i]!=0){
X[j,i]=X[j,i]/(regres$coefficients[1])
Ym[j]=EEX[h,j]/(regres$coefficients[1])
}
if
(Tagestyp[j]==2 && X[j,i]!=0){
X[j,i]=X[j,i]/(regres$coefficients[2])
Ym[j]=EEX[h,j]/(regres$coefficients[2])
}
if
(Tagestyp[j]==3 && X[j,i]!=0){
X[j,i]=X[j,i]/(regres$coefficients[3])
Ym[j]=EEX[h,j]/(regres$coefficients[3])
}
}}

#####
### SVM nur alte Preise
#####

preSVRP=matrix(0,ncol=(length(typesvm)*length(kernelsvm)),nrow=(D-P))
resSVRP=matrix(0,ncol=(length(typesvm)*length(kernelsvm)),nrow=(D-P))
for (i in 0:((D-P)%/I-1)){
svr<-svm(x=X[(1+I*i):(P+I*i)],y=Ym[(1+I*i):(P+I*i)],
kernel=kernelsvm[k],type=typesvm[n],coef=0.25)
preSVRP[(i*I+1):((i+1)*I),pos]<-predict(svr,X[(P+i*I+1):(P+(i+1)*I),])
}

#print(svr)
for (i in 1: length(preSVRP[,pos])){
if (Tagestyp[364+i]==1){
preSVRP[i,pos]=preSVRP[i,pos]*regres$coefficients[1]
}
if (Tagestyp[364+i]==2){
preSVRP[i,pos]=preSVRP[i,pos]*regres$coefficients[2]
}
if (Tagestyp[364+i]==3){

```

```

preSVRP[i,pos]=preSVRP[i,pos]*regres$coefficients[3]
}
}
resSVRP[,pos]<-preSVRP[1:(D-P),pos]-Y[(P+1):D]

#####
### Gewichte EEXm generieren
#####

datareg=as.data.frame(X2[1:365,])
reg<-lm(Y[1:365]~.,datareg)
svr<-svm(x=X2[1:365,],y=Y[1:365],kernel=kernelsvm[k],type=typesvm[n],coef=0.25)
res=svr$fitted-Y[1:365]
res2=res^2
Tagesdummies=matrix(0,nrow=365,ncol=3)
Tagesdummies[,1]=1-X2[1:365,1]
Tagesdummies[,2]=X2[1:365,1]-X2[1:365,2]
Tagesdummies[,3]=X2[1:365,2]

regres<-lm(res2~Tagesdummies[,1]+Tagesdummies[,2]+Tagesdummies[,3]-1)
regres$coefficients=regres$coefficients^(1/2)

for (i in 1:9){
for (j in 1:D){
if (Tagestyp[j]==1 && X2[j,i]!=0){
X2[j,i]=X2[j,i]/(regres$coefficients[1])
Ym[j]=EEX[h,j]/(regres$coefficients[1])
}
if
(Tagestyp[j]==2 && X2[j,i]!=0){
X2[j,i]=X2[j,i]/(regres$coefficients[2])
Ym[j]=EEX[h,j]/(regres$coefficients[2])
}
if
(Tagestyp[j]==3 && X2[j,i]!=0){
X2[j,i]=X2[j,i]/(regres$coefficients[3])
Ym[j]=EEX[h,j]/(regres$coefficients[3])
}
}}

#####
### SVM
#####

preSVR=matrix(0,ncol=(length(typesvm)*length(kernelsvm)),nrow=(D-P))
resSVR=matrix(0,ncol=(length(typesvm)*length(kernelsvm)),nrow=(D-P))
for (i in 0:((D-P)%/%I-2)){
svr<-svm(x=X2[(1+I*i):(P+I*i)],y=Ym[(1+I*i):(P+I*i)],
kernel=kernelsvm[k],type=typesvm[n],coef=0.25)
preSVR[(i*I+1):(i+1)*I,pos]<-predict(svr,X2[(P+i*I+1):(P+(i+1)*I),])

```

```

}
#print(svr)
for (i in 1: length(preSVR[,pos])){
if (Tagestyp[364+i]==1){
preSVR[i,pos]=preSVR[i,pos]*regres$coefficients[1]
}
if
(Tagestyp[364+i]==2){
preSVR[i,pos]=preSVR[i,pos]*regres$coefficients[2]
}
if
(Tagestyp[364+i]==3){
preSVR[i,pos]=preSVR[i,pos]*regres$coefficients[3]
}
}
resSVR[,pos]<-preSVR[,pos]-Y[(P+1):D]

#####
### Prognosefehler
#####

if (n==1){
c=0
}
if (n==2){
c=4
}
###Mittlerer Quadratischer Fehler
MSE[(pos+c),h]=sum(resSVRP[,pos]^2)*(1/length(resSVRP[,pos]))
MSE[(pos+8+c),h]=sum(resSVR[,pos]^2)*(1/length(resSVR[,pos]))
###Mittlerer Absoluter Fehler
MAE[(pos+c),h]=sum(abs(resSVRP[,pos]))*(1/length(resSVRP[,pos]))
MAE[(pos+8+c),h]=sum(abs(resSVR[,pos]))*(1/length(resSVR[,pos]))
###Hitrate
for (i in (P+1):(D-4)){

if (preSVRP[i-P,pos]-Y[i-1]>0 && Y[i]-Y[i-1]>0 || pre
SVRP[i-P,pos]-Y[i-1]<0 && Y[i]-Y[i-1]<0){
Hitrate[(pos+c),h]=Hitrate[(pos+c),h]+1

}

if (preSVR[i-P,pos]-Y[i-1]>0 && Y[i]-Y[i-1]>0 || pre
SVR[i-P,pos]-Y[i-1]<0 && Y[i]-Y[i-1]<0){
Hitrate[(pos+8+c),h]=Hitrate[(pos+c+8),h]+1

}
}
}
print(h)
}
}

```

```
}###Für h Schleife

setwd("C:/Users/Tina/Documents/DA/ErgebnisseR")
unlink("PrognoseSVRG.csv")      ### Altes File löschen
write.csv2(kernelsvm,"PrognoseSVRG.csv",append=TRUE)
write.csv2(typesvm,"PrognoseSVRG.csv",append=TRUE)
write.csv2("MAE","PrognoseSVRG.csv",append=TRUE)
write.csv2(MAE,"PrognoseSVRG.csv",append=TRUE)
write.csv2("MSE","PrognoseSVRG.csv",append=TRUE)
write.csv2(MSE,"PrognoseSVRG.csv",append=TRUE)
Hitrate=Hitrate/((D-4)-(P+1))
write.csv2("Hitrate","PrognoseSVRG.csv",append=TRUE)
write.csv2(Hitrate,"PrognoseSVRG.csv",append=TRUE)
```

```
%
```